

Integració d'una unitat inercial i d'un giroscopi de
fibra òptica en un robot submarí i desenvolupament
d'un sistema de posicionament dins una piscina

Enric Galceran

3 de setembre de 2008

Índex

1	Introducció	3
1.1	Motivació i abast	3
1.2	Objectius	4
1.3	Planificació	4
1.3.1	Organització	4
1.4	Estructura de la memòria	6
2	El robot Ictineu^{AUV}	7
2.1	Descripció del robot	7
2.2	Actuadors	7
2.3	Sensors	8
2.4	Sistema elèctric	8
2.5	Sistema informàtic	9
2.6	Sistemes de coordenades	10
3	L'arquitectura de control O²CA²	11
3.1	Disseny de l'arquitectura	11
3.1.1	Mòdul d'interfície de robot	11
3.1.2	Mòdul de percepció	12
3.1.3	Mòdul de control	13
3.2	Implementació	13
3.2.1	Computació distribuïda amb CORBA	13
3.3	Desenvolupament d'un objecte CORBA d'exemple	15
3.3.1	Desenvolupament de l'objecte Messenger	16

<i>ÍNDEX</i>	2
3.3.2 Proves d'execució de l'objecte Messenger	22
4 Sensors giroscòpics	23
4.1 Antecedents i definició del problema	23
4.2 Plantejament de la solució	24
5 Estudi dels sensors iGC i iFG de Tritech	25
5.1 El sensor iGC (Intelligent Gyro Compass	26
5.1.1 Principis físics	26
5.1.2 Característiques	27
5.2 El sensor iFG (Intelligent Fibre-optic Gyro	28
5.2.1 Principis físics	28
5.2.2 Característiques	29
5.3 Els sensors iGC i iFG com a conjunt sensorial	30
5.3.1 Procés de sincronització	30
5.3.2 Interconnexió dels dispositius	31
5.3.3 Modes d'operació i formats de sortida	32
5.3.4 Calibració del sistema	33
5.4 Experiments previs a la integració dels sensors a l'Ictineu ^{AUV}	33
5.4.1 Anàlisi de la sortida del sistema de sensors	34
5.4.2 Anàlisi de l'efecte provocat pels camps magnètics aliens a la mag- netosfera terrestre	36
5.4.3 Anàlisi de la deriva del sistema	36
5.4.4 Conclusions de l'estudi dels sensors	37
6 Integració dels sensors iGC i iFG a l'Ictineu^{AUV}	40
6.1 Integració dels sensors a l'estructura del robot	40
6.1.1 Instal·lació elèctrica	40
6.1.2 Instal·lació mecànica	41
6.2 Integració dels sensors a l'arquitectura de control O2CA2	43
6.2.1 Desenvolupament de l'aplicació en C++	43
6.2.2 Desenvolupament de l'objecte CORBA per a l'arquitectura O2CA2	46

6.2.3	Desenvolupament d'una interfície gràfica per a visualitzar les dades en temps real	48
6.3	Experiments amb els sensors integrats a l'Ictineu ^{AUV}	51
7	Localització: definició del problema	54
7.1	Antecedents	54
7.2	Plantejament de la solució	56
8	Estudi del sonar MiniKing de Tritech	57
8.1	Interpretació de les imatges capturades pel sonar	58
8.2	Paràmetres de configuració	59
8.3	Desenvolupament d'una interfície gràfica per a la visualització de les imatges acústiques	60
9	Disseny del sistema de localització	61
9.1	Entorn de treball	61
9.2	Disseny de l'algorisme	62
9.2.1	Idea general	62
9.2.2	Sistemes de coordenades	63
9.2.3	Configuració del sonar	64
9.2.4	Segmentació de les imatges acústiques	65
9.2.5	Càlcul de la distància i actualització de la posició	65
9.3	Implementació	67
9.4	Disseny i realització dels experiments de precisió	68
9.5	Resultats	68
10	Conclusions i treballs futurs	73
10.1	Conclusions	74
10.2	Treballs futurs	76

Agraïments

Primer de tot, voldria agrair al meu director del projecte, Marc Carreras, el suport i consells oferts durant tota la realització d'aquest P/TFC. Tot seguit, vull destacar la inestimable ajuda oferta per els meus companys del laboratori: l'Emili Hernández, en Narcís Palomeras, l'Andrés El-Fakdi, en David Ribas i en Lluís Magí. A l'Emili i en Narcís, per haver-me guiat en les fases d'estudi de l'arquitectura de control del robot i a l'hora d'afrontar nous reptes d'aprenentatge. A l'Andrés i en Lluís, per haver treballat sense pausa per instal·lar els sensors giroscòpics a l'estructura de l'Ictineu. I a en David, per les magistrals lliçons de dibuix per ordinador rebudes. Sense la seva aportació i ajuda, que hi ha sigut en tot moment, aquest projecte no hagués estat possible.

Sobretot, vull fer palès l'agradable ambient de treball que es respira al Centre d'Investigació en Robòtica Submarina que ha permès alleugerir les fases més dures de la feina.

I també vull fer una menció especial a tots aquells que, potser sense saber-ho, han contribuït a aportar-me els ànims necessaris per a acabar tot aquest projecte.

Moltes gràcies a tots i a totes.

Capítol 1

Introducció

El Grup de Visió per Computador i Robòtica (VICOROB) del departament d'Electrònica, Informàtica i Automàtica de la Universitat de Girona investiga en el camp de la robòtica submarina. Des de 1992, any en que s'inicià l'activitat investigadora, s'han dut a terme diversos projectes d'investigació que han donat lloc a la construcció de fins a tres vehicles submarins. El més recent, anomenat Ictineu^{AUV} (AUV és l'acrònim de *Autonomous Underwater Vehicle*, vehicle submarí autònom), fou construït l'any 2006 i és actualment l'eina de desenvolupament principal en els projectes d'investigació que el grup VICOROB duu a terme en l'àrea de la robòtica submarina.

1.1 Motivació i abast

Tot i que eventualment es realitzen experiments en entorns naturals allunyats de la universitat, la gran majoria de missions on intervé l'Ictineu^{AUV} es fan a la piscina que es disposa al CIRS (Centre d'Investigació en Robòtica Submarina), laboratori d'investigació que forma part del grup VICOROB.

De cara a l'efectivitat dels sistemes de navegació, peces clau per a la correcta execució de les missions que duu a terme un robot, el fet que aquestes es realitzin en una piscina presenta dues dificultats principals. D'una banda, els materials ferromagnètics presents en el formigó amb que estan construïts els murs de la piscina provoquen distorsions en les mesures de les brúixoles que permeten conèixer l'orientació del robot. De l'altra, les parets llises representen una gran dificultat per al correcte funcionament dels sensors basats en ones acústiques, ja que aquestes reboten fàcilment en les superfícies de la piscina i donen lloc a mesures errònies.

Aquest projecte pretén donar solució a aquestes dues qüestions. Per tant, el projecte s'estructura en dues parts ben diferenciades:

- La primera part adreçarà el problema de la distorsió en les brúixoles mitjançant

l'estudi i la integració d'un nou conjunt de sensors. Aquest nou dispositiu ha estat adquirit recentment pel CIRS i està basat en un giroscopi de fibra òptica que és immune a les distorsions magnètiques que s'han comentat.

- La segona part vol donar solució al problema dels sensors acústics a partir del disseny d'un nou sistema de localització. Aquest sistema es basarà en un sonar i en l'esmentat giròscop de fibra òptica (aprofitant l'alta precisió que proporciona). Estarà pensat per operar tenint en compte les particularitats d'un entorn estructurat com és la piscina del CIRS.

1.2 Objectius

Els objectius d'aquest projecte són els següents:

- Estudiar el funcionament d'un conjunt sensorial basat en una unitat de referència inercial i un giroscopi de fibra òptica.
- Estudiar l'arquitectura de control del robot Ictineu^{AUV} i comprendre el seu funcionament amb la finalitat de construir peces de programari que n'ampliïn la funcionalitat.
- Desenvolupar el programari necessari per integrar aquests sensors a l'arquitectura de control del robot Ictineu^{AUV}.
- Comprovar el correcte funcionament i la precisió del conjunt sensorial mitjançant experimentació.
- Estudiar el funcionament del sonar integrat al robot Ictineu^{AUV}.
- Dissenyar un sistema de localització basat en aquest sonar i en el giròscop de fibra òptica que permeti determinar la posició del robot a la piscina en temps real.
- Implementar aquest sistema de localització.
- Comprovar el correcte funcionament i la precisió del sistema de localització desenvolupat.

1.3 Planificació

1.3.1 Organització

El desenvolupament d'aquest projecte s'ha planificat dividint la feina en les fases que es descriuen a continuació en ordre cronològic.

1. Presa de contacte amb el nou sistema de sensors.

El projecte comença amb la lectura de la documentació que acompanya els nous sensors, arribats al laboratori a principis de desembre de 2007. En aquesta mateixa fase també es realitzen experiments per comprovar les característiques dels sensors, i s'estableix contacte amb l'empresa Tritech per aclarir dubtes sorgits respecte als aparells.

2. Desenvolupament d'una aplicació en C++ per llegir les dades dels sensors.

Un cop estudiat el funcionament de la unitat inercial i del giròscop de fibra òptica, es procedeix a desenvolupar una aplicació que permeti processar les cadenes de text que el sistema de sensors envia cap a un PC. El codi d'aquesta mateixa aplicació serà reutilitzat a l'hora d'integrar els sensors a l'arquitectura de control del robot.

3. Introducció a l'arquitectura de control O2CA2.

El següent pas consisteix en estudiar el disseny del programari que controla el robot Ictineu^{AUV}, on es vol integrar els nous sensors. Aquest procés es duu a terme llegint documentació i desenvolupant programari d'exemple amb la finalitat de familiaritzar-se amb les eines de desenvolupament utilitzades.

4. Desenvolupament del programari necessari per integrar els sensors a l'arquitectura de control del robot.

Un cop estudiats els sensors i el disseny de l'arquitectura de control, es desenvolupa la peça de programari necessària per a integrar els nous sensors. Aquest programari permetrà millorar el sistema de navegació del robot utilitzant l'alta precisió que proporciona el nou sistema de sensors.

5. Disseny i desenvolupament d'una interfície gràfica per a visualitzar les dades provinents dels sensors.

Amb els sensors integrats al sistema de control del robot, es construeix una aplicació GUI que permeti visualitzar en temps real les mesures dels sensors. Fins al moment, no es disposava de cap manera de veure les dades dels sensor en temps real sinó que calia desar les dades en un fitxer durant la missió per analitzar-les després *off-line*.

6. Estudi del sonar integrat a l'Ictineu^{AUV}.

En aquesta fase s'estudia el funcionament del sonar del robot a fi de poder utilitzar-lo en el disseny del sistema de posicionament que ens hem proposat. També es desenvoluparà una aplicació gràfica que permeti visualitzar les imatges acústiques proporcionades pel sonar.

7. Disseny del sistema de posicionament.

Coneixent el funcionament dels sensors giroscòpics i del sonar, en aquesta fase

es dissenya un algorisme que proporcioni la posició del robot dins la piscina del laboratori en temps real.

8. Experimentació amb el sistema de navegació desenvolupat i anàlisi de resultats.

Un cop desenvolupat el sistema de navegació, es realitzen experiments per determinar-ne la precisió i fiabilitat.

9. Documentació.

Finalment, s'elabora la documentació necessària per mostrar el procés de realització del projecte.

1.4 Estructura de la memòria

Aquest document s'estructura de la següent manera. En els capítols 1, 2 i 3 es fa una presentació del projecte i s'explica el marc on es desenvolupa i les eines que s'utilitzen per a dur-lo a terme. Els capítols del 4, 5, 6, expliquen com s'ha integrat els sensors giroscòpics al robot Ictineu^{AUV} a fi de poder disposar d'una mesura d'orientació relativa precisa i lliure de distorsions magnètiques. En els capítols 7, 8, 9 s'explica el procés de disseny del sistema de posicionament basat en un giròscop de fibra òptica i un sonar i els experiments realitzats per determinar-ne la fiabilitat i la precisió. Finalment, en el capítol 10 s'explica les conclusions obtingudes del treball realitzat s'exposen els treballs futurs que poden ser interessants de desenvolupar a partir de la feina feta en aquest projecte.

Capítol 2

El robot Ictineu^{AUV}

El robot Ictineu^{AUV} ha estat el banc de proves on s'ha desenvolupat aquest projecte. Fou construït durant el primer semestre de 2006 per un equip d'estudiants de la UdG amb la finalitat d'afrontar el repte del SAUC-E'06 (*Student Autonomous Underwater Challenge - Europe*), la primera competició de robòtica submarina disputada a Europa. Per els requeriments de la competició, es va voler construir un vehicle d'alta manio-bravilitat mantenint una estructura simple que en facilités el manteniment i futures actualitzacions.

2.1 Descripció del robot

L'estructura del robot, un disseny habitualment utilitzat en robots comercials, té les següents dimensions: 0.74 m. de llarg, 0.47 m. d'alt i una amplada de 0.53 m. (vegeu la figura 2.1). Aquesta estructura és feta de polioximetilè, un polímer altrament conegut amb el nom comercial de Delrin. La flotabilitat necessària per als seus 53 Kg. de pes és proporcionada per una coberta d'escuma amb un volum de 10.5 litres.

2.2 Actuadors

El vehicle submarí és propulsat per sis motors (vegeu la figura 2.2). Dos d'ells, situats verticalment, permeten al robot moure's en *heave* (moviment en l'eix Z). Els altres quatre motors estan situats horitzontalment i la seva orientació estratègica a 45° de l'eix del vehicle permeten al robot moure's tant en *surge* (moviment en l'eix X) com en *sway* (moviment en l'eix Y), a la vegada que li permeten variar la seva orientació en el pla horitzontal (*heading* o *yaw*).



Figura 2.1: El robot Ictineu^{AUV} operant a la piscina del CIRS.

2.3 Sensors

Al moment d'iniciar aquest projecte l'Ictineu^{AUV} estava equipat amb un conjunt de sensors format per una càmera a color de visió frontal, una càmera en blanc i negre orientada cap al fons, una unitat de referència inercial model MTi de Xsens Technologies per conèixer la orientació relativa del vehicle, un sònar de formació d'imatges (*Mechanically Scanned Imaging Sonar* - MSIS) model MiniKing de Tritech, una ecosonda d'Airmar i un DVL (*Argonaut Doppler Velocity Logger*) de Sontek, dispositiu basat en l'efecte Doppler que permet mesurar la velocitat del vehicle i que inclou una brúixola, inclinòmetres, un sensor de pressió i un altímetre.

D'aquest conjunt sensorial inicialment present al robot, bàsicament només el sonar serà d'interès per al desenvolupament d'aquest projecte, ja que el nou sensor giroscòpic substituirà la unitat de referència inercial MTi.

2.4 Sistema elèctric

El sistema elèctric es compon de sis controladors de potència per als motors així com de dues bateries de plom que s'han dimensionat per a obtenir una autonomia de dues hores. A més, el vehicle pot ser alimentat amb potència externa mitjançant un cable, anomenat umbilical, que permet més estona de treball. Aquests dos modes de treball es commuten amb un circuit de relés.

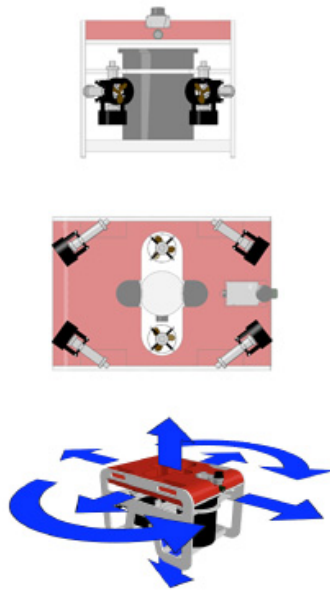


Figura 2.2: Esquema de la distribució dels propulsors del robot i de les forces que exerceixen.

2.5 Sistema informàtic

Els components electrònics del robot s'allotgen en dos cilindres resistents a la pressió exercida per l'aigua fets d'alumini. Un dels cilindres conté els ordinadors utilitzats per al control del sistema, i l'altre conté bateries i electrònica diversa.

El sistema informàtic del robot consta de dos ordinadors PC connectats en xarxa LAN mitjançant un router. Inicialment, el conjunt es va dissenyar perquè un dels ordinadors s'encarregués de les tasques de visió per computador, que requereixen més temps de processador, i l'altre s'encarregués del control automàtic. Ara mateix, però aquesta divisió és una mica ambigua ja que les tasques de visió i control es reparteixen entre els dos ordinadors més o menys equitativament. Tot i això, un dels ordinadors utilitza el hostname control i l'altre el hostname visio.

El router que connecta els dos ordinadors del robot es connecta també, a través d'un cable anomenat umbilical, a la xarxa informàtica del laboratori. Això permet utilitzar els PC del robot com si es tractés de dues estacions de xarxa més, facilitant la interacció amb el sistema.

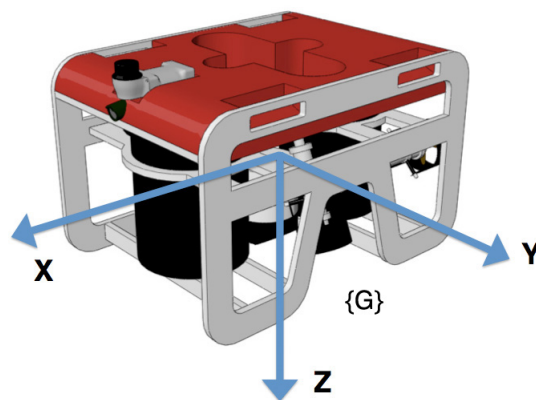
Posició	Velocitat	Grau de llibertat
θ	p	<i>Roll</i> , rotació respecte l'eix X
ϕ	q	<i>Pitch</i> , rotació respecte l'eix Y
Ψ	r	<i>Yaw</i> , rotació respecte l'eix Z
X	u	Posició en l'eix X o <i>surge</i>
Y	v	Posició en l'eix Y o <i>sway</i>
Z	w	Posició en l'eix Z o <i>heave</i>

Taula 2.1: Graus de llibertat d'un robot submarí.

2.6 Sistemes de coordenades

En robòtica submarina, per determinar la posició d'un vehicle cal definir dos sistemes de coordenades. Un d'ells es situa arbitràriament en un punt conegut de la Terra i s'anomena sistema de coordenades de món $\{E\}$. L'altre, anomenat sistema de coordenades de robot $\{G\}$, es situa convenientment sobre el vehicle, generalment al seu centre de gravetat i amb els eixos tal i com es mostra a la figura 2.3: l'eix X sortint des del morro del vehicle, l'eix Y perpendicular al primer en el pla horitzontal i l'eix Z en direcció vertical i amb el sentit positiu cap avall.

La taula 2.1 mostra els sis graus de llibertat en els quals pot moure's un robot submarí. L'Ictineu^{AUV} realitza desplaçaments en només quatre d'aquests graus de llibertat. Aquests són els moviments lineals al llarg dels eixos del sistema de coordenades (moviment en X o *surge*, moviment en Y o *sway* i moviment en Z o *heave*) i la rotació només sobre l'eix Z (*yaw*, també anomenat *heading*), ja que el robot és passivament estable pel que fa a les rotacions en els eixos X (*roll*) i Y (*pitch*).

Figura 2.3: Sisema de coordenades de robot $\{G\}$.

Capítol 3

L'arquitectura de control O²CA²

Per a realitzar el control automàtic d'un robot com l'Ictineu^{AUV} cal utilitzar un sistema de control complex que respongui als requeriments d'un vehicle autònom. Per aquesta raó, les missions que realitza l'Ictineu^{AUV} es controlen amb el que s'anomena una arquitectura de control: un programari que, bàsicament, processa la informació que prové dels sensors i d'acord amb aquesta envia consignes als actuadors del robot.

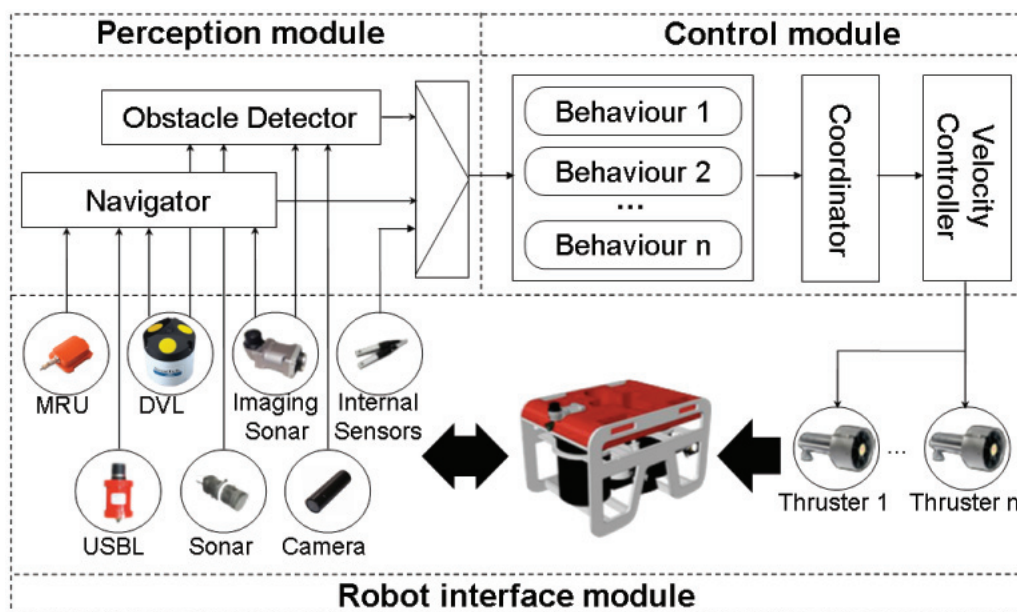
Per assolir els objectius plantejats en aquest projecte és indispensable conèixer bé aquesta arquitectura de control, ja que la integració tant d'un nou conjunt sensorial com d'un sistema de posicionament a l'Ictineu^{AUV} impliquen el disseny i desenvolupament de peces de programari que passaran a formar-ne part.

3.1 Disseny de l'arquitectura

L'arquitectura de control utilitzada en l'Ictineu^{AUV} s'anomena O²CA² (*Object-Oriented Control-Architecture for Autonomy*). Es tracta d'una arquitectura multirobot, dissenyada per a poder ser utilitzada en diferents sistemes robòtics. L'arquitectura de control es divideix en tres mòduls: el mòdul d'interfície del robot, el mòdul de percepció i el mòdul de control.

3.1.1 Mòdul d'interfície de robot

Aquest és l'únic mòdul que conté objectes que interactuen amb el maquinari directament. Està format bàsicament per dos tipus d'objectes: objectes sensor responsables de llegir les dades captades pels sensors físics i objectes actuator encarregats d'enviar comandes als actuadors del robot. Els objectes sensor de l'Ictineu^{AUV} inclouen un DVL, un MSIS, una MRU, dues cameres, un sensor de profunditat i una ecosonda. Tamé hi ha objectes encarregats de monitoritzar la pressió, presència d'aigua i temperatura interna

Figura 3.1: Esquema de l'arquitectura O²CA²

del sistema. Els objectes actuator consisteixen en els sis motors.

3.1.2 Mòdul de percepció

Aquest mòdul consta de dos components bàsics: el Navegador i el Detector d'Obstacles, que són bàsicament sensors virtuals. Un sensor virtual és un programari que produeix informació sensorial o bé a partir d'un simulador de missions o bé a partir de la conjunció de dades provinents de diversos sensors físics. L'objecte Navegador s'encarrega d'estimar la posició, velocitat i acceleració del robot. Per acomplir aquesta tasca, existeix una interfície anomenada *NavigationSensor* de la qual els sensors de localització (DVL, MRU, sensor de profunditat) hereden. Aquesta interfície proporciona a aquests sensors un conjunt de mètodes per a retornar la posició, velocitat i acceleració en sis graus de llibertat juntament amb una estimació de la qualitat d'aquestes mesures. Actualment, el Navegador estima la posició (x, y, z) , la orientació $(roll, pitch, yaw)$ i l'acceleció lineal i les velocitats angulars.

L'objecte Detector d'Obstacles s'utilitza per detectar objectes presents al fons mitjançant tres distàncies proporcionades pel DVL. Properament, però, s'utilitzarà també per detectar obstacles frontals utilitzant l'ecosonda i/o el sonar.

Així, el mòdul de percepció utilitza la navegació i les dades de detecció d'obstacles, essent independent dels sensors físics.

3.1.3 Mòdul de control

El controlador d'alt nivell és la part més complexa de l'arquitectura. Pel que concerneix a aquest projecte, sols cal tenir en compte que el control d'alt nivell es realitza a partir d'uns programes informàtics anomenats comportaments dissenyats per a assolir un objectiu concret. Exemples típics de comportaments poden ser el comportament mantenir l'altitud o bé mantenir la orientació. Les respostes generades individualment per cada comportament són coordinades d'acord amb un conjunt de prioritats i directives per generar les consignes que s'envien al controlador de baix nivell.

3.2 Implementació

La implementació de l'arquitectura O²CA² segueix el paradigma de la programació orientada a objectes: un conjunt d'elements o objectes (corresponents als sensors, als controladors, etc...) interactuen entre sí. Per a desenvolupar el programari es va escollir el llenguatge de programació C++ i el sistema operatiu GNU/Linux.

Però, com s'ha comentat al capítol 2, el sistema informàtic de l'Ictineu^{AUV} consta de dos PC connectats en xarxa, i és per tant un sistema distribuït. L'arquitectura O²CA² s'ha implementat de manera que els seus objectes puguin interactuar transparentment a través de la xarxa com si residissin en el mateix sistema.

3.2.1 Computació distribuïda amb CORBA

Per a permetre al programari de l'arquitectura operar en sistemes distribuïts aprofitant-ne eficientment les característiques, la implementació es va realitzar utilitzant l'estàndard CORBA (*Common Object Request Broker Architecture* - arquitectura comuna d'intermediaris en peticions a objectes), que estableix una plataforma de desenvolupament de sistemes distribuïts facilitant la invocació de mètodes remots sota un paradigma orientat a objectes.

Essencialment, el que permet CORBA és utilitzar tot un sistema distribuït com si es tractés d'un sol computador. Els objectes CORBA poden interactuar entre sí efectuant crides als seus mètodes independentment de l'ordinador de la xarxa on s'estan executant.

CORBA és un estàndard, un conjunt de propostes definit per un consorci d'empreses de la indústria informàtica anomenat OMG (*Object Management Group*). Aquest estàndard ha donat lloc a varies implementacions pensades per utilitzar-se en diferents sistemes operatius i amb diversos llenguatges de programació.

Concretament, l'arquitectura O²CA² s'ha desenvolupat utilitzant TAO (*The ACE ORB*), una implementació de CORBA per a C++ que compleix els estàndards i pensada per a aplicacions de temps real. El sistema operatiu utilitzat per al desenvolupament és

GNU/Linux.

Conceptes i terminologia

Com totes les tecnologies, CORBA té una terminologia única associada a ella. Tot i que alguns dels seus conceptes han estat presos de tecnologies similars, molts altres són nous o diferents. És per això cal tenir clar i saber a què es refereix cada un dels següents termes quan es parla en el context de CORBA:

- Un objecte CORBA (CORBA object) és una entitat virtual capaç de ser localitzada mitjançant un ORB i d'atendre peticions invocades en ella. És virtual en el sentit que no existeix realment excepte si és concretada per una implementació feta en un llenguatge de programació.
- Un objecte destí (target object), en el context d'una petició d'invocació CORBA, és l'objecte CORBA destí d'aquesta petició.
- Un client és una entitat que invoca una petició en un objecte CORBA. El terme client té sentit només en el context d'una petició particular, ja que l'aplicació que és client per una petició pot ser el servidor per a una altra petició.
- Un servidor és una aplicació on existeixen un o més objectes CORBA. Com en els clients, el terme té sentit només per a una petició en concret.
- Una petició (request) és la invocació d'una operació en un objecte CORBA per part d'un client.
- Una referència a un objecte (object reference) és un manipulador utilitzat per identificar, localitzar i adreçar un objecte CORBA.

Realment aquests conceptes són molt abstractes, però prenen sentit quan es programa utilitzant CORBA i és molt important tenir-los en compte.

El servidor de noms o Naming Service

Per permetre als objectes interactuar a través de la xarxa, CORBA utilitza el que s'anomena un Naming Service. El Naming Service és un servidor de noms, és a dir, una aplicació que s'executa en un dels ordinadors de la xarxa que els objectes utilitzen per a identificar-s'hi. De fet la seva funció és similar al protocol DNS utilitzat a Internet, però cal no confondre'ls. Mentre que el protocol DNS permet obtenir adreces IP a partir de noms de domini, el Naming Service permet obtenir referències a objectes mitjançant un nom.

La figura 3.2 mostra el funcionament del Naming Service:

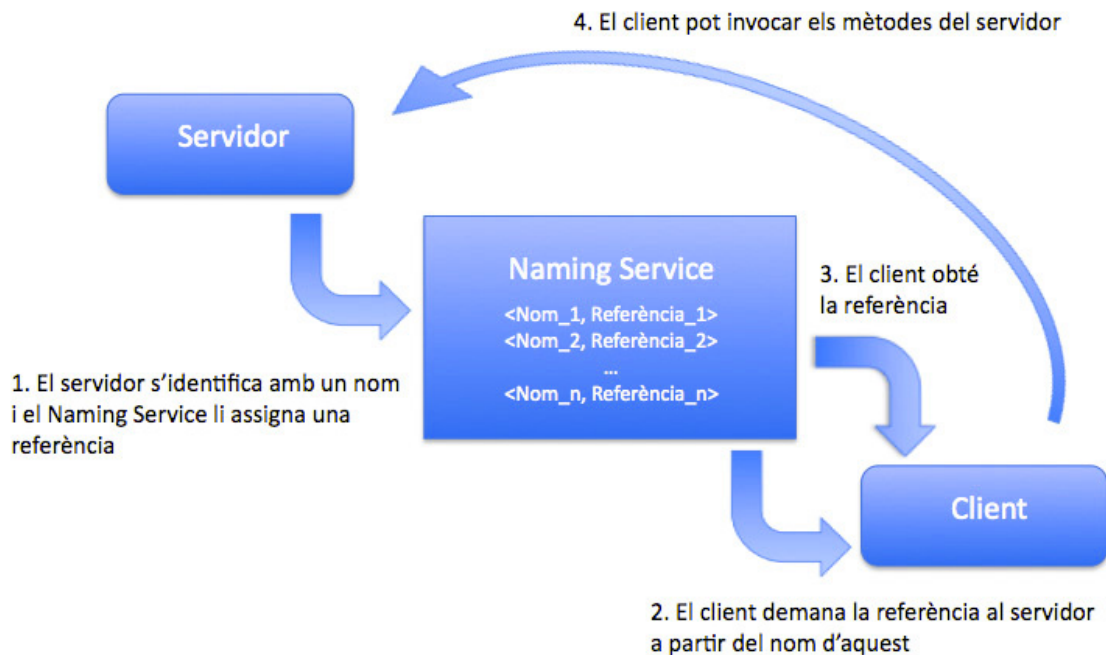


Figura 3.2: Esquema del funcionament del Naming Service

1. En iniciar-se, els objectes es registren al l'espai de noms del Naming Service fent servir un nom únic, determinat pel programador.
2. Quan un objecte client vol realitzar una petició a un altre objecte, obté una referència a partir del nom de l'objecte destí.
3. El Naming Service retorna al client una referència a l'objecte sol·licitat.
4. Un cop obtinguda la referència, el client ja pot realitzar peticions a l'objecte destí.

3.3 Desenvolupament d'un objecte CORBA d'exemple

Per aconseguir la tasca d'integrar un nou sensor a l'arquitectura de control caldrà conèixer com dissenyar i desenvolupar un objecte CORBA utilitzant les tecnologies anteriorment descrites. Tot seguit es mostra el procés seguit per desenvolupar un objecte d'exemple amb la finalitat de familiaritzar-se amb l'arquitectura. Aquesta tasca fou duta a terme durant el mes de desembre de 2007.

El procés de desenvolupament consisteix en els següents passos. Primerament, es defineix la interfície de l'objecte amb els seus mètodes, atributs i estructures de dades

que utilitzarà mitjançant el llenguatge IDL (*Interface Definition Language*). Aquest llenguatge és independent de la implementació concreta de CORBA que s'està utilitzant i també del llenguatge de programació. Un cop definit l'objecte, es compila el codi en IDL i es genera un esquelet en el llenguatge corresponent a la implementació utilitzada, C++ en el cas que ens ocupa. L'esquelet generat conté només la signatura dels mètodes definits, i cal escriure'n el codi font. Un cop fet això, ja pot compilar-se l'objecte per a obtenir els fitxers binaris que s'executaran.

3.3.1 Desenvolupament de l'objecte Messenger

L'exemple que es desenvoluparà es proposa al document anomenat *Getting Started* que acompanya el TAO. Consisteix en un senzill objecte que consta d'un sol mètode. Aquest mètode, en ser executat, mostra per pantalla el missatge que se li ha passat per paràmetre.

L'objecte s'ha desenvolupat fent servir les eines de disseny i programació utilitzades al laboratori de robòtica submarina on s'ha treballat, és a dir: una estació de treball amb GNU/Linux, el compilador de C++ gcc i la distribució de CORBA TAO en la seva versió 1.3. A continuació es detalla el procés de desenvolupament pas a pas.

1. **Crear el directori de treball.** Obrim un terminal de consola i teclegem les següents comandes:

```
mkdir ./Messenger
cd Messenger
```

2. **Definir la interfície de l'objecte amb el llenguatge IDL.** Crearem un fitxer anomenat *Messenger.idl* on definirem la interfície de l'objecte. En aquesta interfície només es defineixen els atributs i mètodes públics de l'objecte, no els privats. L'objecte Messenger, com s'ha comentat, consta d'un sol mètode anomenat *sendmessage* que mostra per pantalla el missatge que se li ha passat per paràmetre. Veiem tot seguit el contingut del fitxer *Messenger.idl*:

```
// Messenger.idl

interface Messenger
{
    boolean send_message(in string user_name,
                        in string subject,
                        inout string message);
};
```

3. **Compilar el fitxer IDL.** A continuació es compila el fitxer IDL per a generar l'esquelet en llenguatge C++.

```
tao_idl -GI Messenger.idl
```

Aquesta comanda genera els següents fitxers:

- *MessengerS.h*
- *MessengerS.cpp*
- *MessengerC.h*
- *MessengerC.cpp*
- *MessengerI.h*
- *MessengerI.cpp*

De tots aquests fitxers només són interessants els anomenats *MessengerI.h* i *MessengerI.cpp*, que són els que contenen l'esquelet de l'objecte. La resta de fitxers contenen el codi necessari per tal que es pugui accedir a l'objecte transparentment en un sistema distribuït mitjançant CORBA, i no cal modificar-los.

4. **Crear els fitxers d'implementació.** És recomanable canviar el nom dels fitxers generats pel compilador d'IDL, *MessengerI.h* i *MessengerI.cpp*, per *Messenger i.h* i *Messenger i.cpp*, respectivament. Això evitarà que es sobreescriguin les modificacions que s'hagin fet a l'esquelet de l'objecte si es torna a executar el compilador d'IDL.

```
mv MessengerI.h Messenger i.h
mv messengerI.cpp Messenger i.cpp
```

5. **Completar el contingut del fitxer Messenger i.h.** El fitxer de capçalera *Messenger i.h* conté les definicions dels mètodes públics de l'objecte, i hi podem afegir atributs i mètodes privats i modificar, si escau, alguna de les definicions generades automàticament pel compilador d'IDL. Pel que fa a aquest exemple tant senzill, no caldrà fer-hi cap modificació.

```
/* -*- C++ -*- $Id: Messenger_i.h,v 1.1.2.2 2003/07/14 21:59:20
   michel_j Exp $ */

// ***** Code generated by the The ACE ORB (TAO) IDL Compiler
// *****
// TAO and the TAO IDL Compiler have been developed by the Center for
// Distributed Object Computing at Washington University, St. Louis.
//
// Information about TAO is available at:
// http://www.cs.wustl.edu/~schmidt/TAO.html

#ifndef MESSENGER_I_H_
#define MESSENGER_I_H_

#include "MessengerS.h"

#if !defined (ACELACKS_PRAGMA_ONCE)
#pragma once
```



```

#endif /* ACELACKS_PRAGMA_ONCE */

//Class Messenger_i
class Messenger_i : public virtual POA_Messenger
{
public:
    //Constructor
    Messenger_i (void);

    //Destructor
    virtual ~Messenger_i (void);

virtual CORBA::Boolean send_message (
    const char * user_name,
    const char * subject,
    char *& message
    )
    ACE_THROW_SPEC((CORBA::SystemException));

};

#endif /* MESSENGERLH_ */

```

6. **Completar el contingut del fitxer Messenger i.cpp.** Al fitxer *Messenger i.cpp* hi ha la implementació dels mètodes definits al fitxer de capçalera. El modificarem per fer que el mètode send message mostri la informació que se li ha passat per paràmetre en ser cridat.

```

#include "started_pch.h"
/* -*- C++ -*- $Id: Messenger_i.cpp,v 1.1.2.2 2003/07/14 21:59:20
    michel_j Exp $ */

// ***** Code generated by the The ACE ORB (TAO) IDL Compiler
// *****
// TAO and the TAO IDL Compiler have been developed by the Center for
// Distributed Object Computing at Washington University, St. Louis.
//
// Information about TAO is available at:
//      http://www.cs.wustl.edu/~schmidt/TAO.html

#include "Messenger_i.h"
#include <ace/streams.h>

// Implementation skeleton constructor
Messenger_i::Messenger_i (void)
{
}

// Implementation skeleton destructor
Messenger_i::~~Messenger_i (void)

```

```

{
}

CORBA::Boolean Messenger_i::send_message (
    const char * user_name,
    const char * subject,
    char *& message
)
ACE_THROW_SPEC((CORBA::SystemException))

{
    cout << "Message_from:_ " << user_name << endl;
    cout << "Subject:_" << subject << endl;
    cout << "Message:_" << message << endl;
    CORBA::string_free(message);
    message = CORBA::string_dup("Thanks_for_the_message.");
    return 1;
}

```

7. **Implementar el servidor.** El servidor d'un objecte CORBA consisteix en un procés de sistema que espera activament rebre peticions per part dels clients de l'objecte. Quan arriba una petició l'atén executant el mètode de l'objecte demanat, i continua fent això fins que el procés és finalitzat.

D'entrada aquest codi pot semblar molt complex per a un servidor que només escriu un missatge per pantalla, però la majoria d'aquest codi correspon a una plantilla que es repeteix idènticament en tots els servidors CORBA excepte per unes poques línies.

```

#include "started_pch.h"

#include "Messenger_i.h"
#include <ace/streams.h>

int main( int argc, char *argv[] )
{
    try {
        // Initialize the ORB.
        CORBA::ORB_var orb = CORBA::ORB_init( argc, argv );

        //Get reference to the RootPOA.
        CORBA::Object_var obj = orb->resolve_initial_references( "RootPOA" );
        PortableServer::POA_var poa = PortableServer::POA::_narrow( obj.in() );

        // Activate the POAManager.
        PortableServer::POAManager_var mgr = poa->the_POAManager();
        mgr->activate();

        // Create a servant.

```

```

    Messenger_i servant;

    // Register the servant with the RootPOA, obtain its object
    // reference, stringify it, and write it to a file.
    PortableServer::ObjectId_var oid = poa->activate_object( &servant
    );
    obj = poa->id_to_reference( oid.in() );
    CORBA::String_var str = orb->object_to_string( obj.in() );
    ofstream iorFile( "Messenger.ior" );
    iorFile << str.in() << endl;
    iorFile.close();
    cout << "IOR_written_to_file_Messenger.ior" << endl;

    // Accept requests from clients.
    orb->run();
    orb->destroy();

    return 0;
}
catch (CORBA::Exception& ex) {
    cerr << "CORBA_exception:_" << ex << endl;
}

return 1;
}

```

8. **Implementar el client.** Un client CORBA es connecta a l'objecte del tipus i amb el nom especificats mitjançant el Naming Service. Si la connexió es realitza amb èxit, s'obté una referència a l'objecte que permet cridar-ne els mètodes. En aquest cas, el client simplement executa un sol com el mètode send message de l'objecte Messenger.

Anàlogament al cas del servidor, gairebé tot el codi aquí presentat correspon a un patró que es repeteix per a tots els clients CORBA, i del qual només cal modificar unes poques línies.

```

#include "started_pch.h"

#include "MessengerC.h"
#include <ace/streams.h>

int main( int argc, char* argv[] )
{
    try {
        // Initialize the ORB.
        CORBA::ORB_var orb = CORBA::ORB_init( argc, argv );

        // Read and destringify the Messenger object's IOR.
        CORBA::Object_var obj = orb->string_to_object( "file://Messenger.ior" );
        if( CORBA::is_nil( obj.in() ) ) {

```

```

    cerr << "Could_not_get_Messenger_IOR." << endl;
    return 1;
}

// Narrow the IOR to a Messenger object reference.
Messenger_var messenger = Messenger::_narrow( obj.in() );
if( CORBA::is_nil( messenger.in() ) ) {
    cerr << "IOR_was_not_a_Messenger_object_reference." << endl;
    return 1;
}

// Send a message the the Messenger object.
CORBA::String_var msg = CORBA::string_dup( "Hello!" );
messenger->send_message( "TAO_User", "TAO_Test", msg.inout() );

// Print the Messenger's reply.
cout << "Reply:_" << msg.in() << endl;

    return 0;
}
catch ( CORBA::Exception& ex ) {
    cerr << "CORBA_exception:_" << ex << endl;
}

    return 1;
}

```

9. **Crear els Makefiles per al GNU build system.** Tot i que l'objecte que s'està desenvolupant és molt senzill, les eines de desenvolupament del TAO generen un munt de fitxers que cal compilar i enllaçar. El problema es fa encara més difícil quan en el desenvolupament d'un objecte hi intervenen l'herència i les referències a altres objectes CORBA.

Per facilitar la tasca de compilació, el TAO inclou una eina anomenada MPC (*Makefile Project Creator*). Aquesta eina utilitza fitxers MPC on es defineix quins fitxers de codi font s'utilitzen en el projecte que es vol compilar. A partir d'aquests fitxers es generen *makefiles*, els fitxers de compilació automàtica utilitzats per el GNU build system propi dels sistemes UNIX. El principal avantatge d'utilitzar fitxers MPC i no *makefiles* directament per a la compilació és que un projecte MPC es pot portar sense dificultat a un sistema operatiu diferent. Per exemple, en un sistema Windows es generaria un projecte per a Visual Studio, o en MacOS X es generaria un projecte per a Xcode.

```

project(*Server): devguide_server {
    idlflags += -Wb,pch_include=started_pch.h

    Source_Files {
        Messenger_i.cpp
        MessengerServer.cpp
    }
}

```

```

    }
}

project(*Client): devguide_client {
    idlflags += -Wb,pch_include=started_pch.h

    Source_Files {
        MessengerC.cpp
        MessengerClient.cpp
    }
}

```

10. **Compilar el codi.** Un cop hem obtingut els *makefiles* per al client i el servidor del nostre objecte Messenger, procedim a compilar el codi amb la comanda *make*:
- ```
make -f Makefile ...
```

### 3.3.2 Proves d'execució de l'objecte Messenger

Si no hi ha hagut errors en la compilació ni en l'enllaç es pot procedir a executar les aplicacions construïdes. El primer que cal fer és engegar el Naming Service en un terminal:

```
$TAO_ROOT/orbsvc/NamingService/NamingService -m 1
```

A continuació executem l'aplicació MessengerServer en un altre terminal:

```
./MessengerServer
```

El servidor produeix la següent sortida:

```
Message from: TAO User
Subject: Test
Message: Hello!
```

A continuació executem el MessengerClient. Per a fer palesa la capacitat de CORBA per a treballar utilitzant la xarxa podem executar l'aplicació en una altra estació de treball. Utilitzarem la següent comanda:

```
./MessengerClient
```

Produeix la següent sortida:

```
Reply: Thanks for the message.
```

Com s'ha pogut observar, el servidor ha esperat la petició del client. Quan aquesta ha arribat, l'ha atès enviant el missatge al client. Quan el client ha rebut el missatge, ho ha notificat i el programa ha finalitzat. Per contra, el servidor segueix executant-se esperant noves peticions d'altres clients, i seguirà així fins que es finalitzi manualment el procés.

# Capítol 4

## Sensors giroscòpics

En aquest capítol s'expliquen els problemes que sorgeixen quan es vol determinar la orientació relativa d'un vehicle submarí en un entorn estructurat (llegeixi's piscina) mitjançant una unitat inercial basada en la mesura de camps magnètics, tal i com es feia abans de l'adquisició del nou conjunt de sensors que motiva la realització d'aquest projecte. A continuació s'explicarà com es pot donar solució a aquests problemes emprant un giròscop de fibra òptica i els passos que cal seguir a fi d'instal·lar el dispositiu a l'Ictineu<sup>AUV</sup> i disposar d'unes mesures d'orientació precises i lliures de distorsions magnètiques.

### 4.1 Antecedents i definició del problema

Fins el desembre de 2007, data d'inici d'aquest projecte, el CIRS disposava només d'una unitat de referència inercial model MTi de la casa Xsens Technologies per a realitzar mesures de l'orientació en vehicles submarins. Aquesta unitat de referència inercial es basa en dispositius magnètics (brúixoles i inclinòmetres) per donar una mesura de la orientació del sistema respecte el nord magnètic terrestre i de la rotació en pitch i en roll. Com que el sensor es basa en la magnetosfera de la Terra, les seves mesures es veuen afectades per l'acció de camps aliens al del planeta.

Com tota unitat inercial, el sensor MTi pot calibrar-se per tal que compensi l'acció de camps magnètics constants provocats per cossos propers. Per exemple, en el cas del robot Ictineu<sup>AUV</sup>, aquest sensor es trobava rodejat de cables elèctrics, del metall que compon els cilindres del vehicle, d'elements de l'estructura, etc... Sabent que aquests elements es troben sempre en la mateixa posició relativa al sensor i que per tant hi exerceixen un camp magnètic constant, pot calibrar-se la unitat realitzant el que s'anomena un swing calibration procedure.

Aquest procés consisteix en prendre un segon dispositiu de mesura, per exemple una brúixola tradicional, i moure el sensor cap a una orientació determinada per aquest



Figura 4.1: La unitat inercial MTi

segon sistema. Llavors, es mesura la diferència entre el valor que proporciona la unitat inercial i el valor que proporciona el sistema extern. Aquest procés es repeteix per a diversos punts al llarg dels 360° d'una volta. Finalment, a partir dels punts mesurats es construeix una corba de compensació extrapolant entre els punts mesurats.

Tot i que aquest procés permet evitar les distorsions magnètiques constants, no pot evitar les que són provocades per cossos aliens a l'estructura del vehicle. És a dir, en el cas particular de la piscina del laboratori, el sensor es veu afectat per l'estructura de formigó armat de la piscina. Com que aquestes distorsions són provocades per cossos que no es troben sempre en la mateixa posició relativa al sensor, els camps magnètics que provoquen són variables i per tant no és possible calibrar el sensor perquè pugui evitar-los.

En conseqüència, el sensor MTi donava errors absoluts de fins a 20° en la mesura de l'angle de *heading*.

## 4.2 Plantejament de la solució

Per tot el que s'acaba de comentar, es va decidir adquirir un nou sensor basat en un giròscop de fibra òptica. Fent servir aquesta tecnologia, es poden obtenir mesures altament precises de la orientació d'un vehicle que no es distorsionen a causa de la presència de camps magnètics. Així, doncs, el que caldrà fer és estudiar el funcionament d'aquest nou dispositiu i integrar-lo a l'Ictineu<sup>AUV</sup> per tal de proporcionar al vehicle aquestes mesures.

## Capítol 5

# Estudi dels sensors iGC i iFG de Tritech

El nou conjunt sensorial destinat a substituir la unitat de referència inercial MTi consta de dos sensors (mostrats a la figura 5.1) fabricats per la casa Tritech. El primer, anomenat iGC (*Intelligent Gyro Compass*) consisteix en un conjunt de giròscops, magnetòmetres i acceleròmetres que permeten obtenir dades de la orientació relativa en tres graus de llibertat: *roll*, *pitch* i *yaw*. Per construcció, aquest sensor és sensible a les pertorbacions magnètiques de l'entorn on es troba.

El segon sensor, anomenat iFG (*Intelligent Fibre-optic Gyro*) consisteix en un giròscop de fibra òptica que utilitza interferència lumínica per detectar la rotació mecànica d'una bobina de fibra òptica. Aquest sensor proporciona dades sobre la orientació en un sol grau de llibertat (cal dir que es tracta d'un sensor molt més car i complex que un giròscop convencional). Com que l'Ictineu<sup>AUV</sup> es considera passivament estable en *roll* i *pitch*, on serà interessant la funció d'aquest sensor és en la mesura de l'orientació en *yaw*.



Figura 5.1: Els sensors iGC (esquerra) i iFG (dreta)



## 5.1 El sensor iGC (Intelligent Gyro Compass)

### 5.1.1 Principis físics

El sensor iGC es basa en el funcionament d'un aparell anomenat giròscop. El giròscop o giroscopi és un dispositiu mecànic format essencialment per un cos amb simetria de rotació que gira al voltant del seu eix de simetria. Quan és sotmès a un moment de força que tendeix a canviar la orientació de l'eix de rotació el seu comportament és aparentment paradoxal, ja que l'eix de rotació, en lloc de canviar de direcció com ho faria un cos que no girés, canvia d'orientació en una direcció perpendicular al moment aplicat. Així, quan el giròscop no està rotant es comporta com un objecte qualsevol, però quan se'l fa girar al voltant del seu eix a gran velocitat el giròscop es resisteix a moure's en certes direccions.

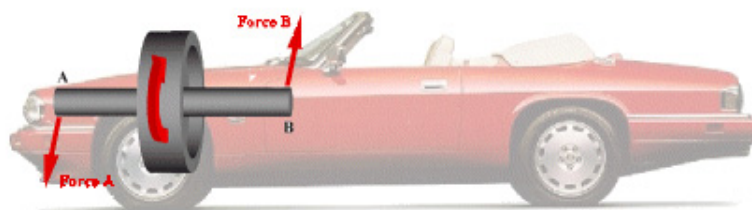


Figura 5.2: Exemple d'un giròscop instal·lat a l'estructura d'un cotxe

Quan el giròscop gira conté grans quantitats d'energia emmagatzemada. La primera llei de Newton diu que qualsevol cos continuarà en el seu estat de moviment (ja sigui quiet o en moviment rectilini uniforme) fins que una força hi actui. A causa de les forces exercides sobre el giròscop quan gira, si se'l mou, el giròscop intentarà *compensar* aquest moviment. Prenem l'exemple d'un giròscop instal·lat longitudinalment a l'estructura d'un cotxe, tal com es mostra a la figura 5.2 (sovint el motor d'un cotxe es comporta com un giròscop degut a la seva forma, massa i moviments de rotació).

Suposem que el giròscop està girant a una velocitat elevada en el sentit de les agulles del rellotge (mirant des del darrere del cotxe i tal com es mostra a la figura). Quan el cotxe gira cap a la dreta (en el sentit de les agulles del rellotge mirant-lo des de dalt) les forces A i B s'apliquen a l'estructura del vehicle forçant que la part frontal (el morro) s'abaixi i que la part posterior (el cul) s'aixequi. Inversament, si el cotxe gira cap a l'esquerra passa el contrari: el morro s'aixeca i el cul s'abaixa.

No obstant, si el cotxe es mou cap endavant o cap endarrera, o lateralment cap a l'esquerra o cap a la dreta (sense girar) el giròscop no exerciria cap força addicional. El giròscop només exerceix forces extra quan el cotxe és rotat en un angle: o bé girant cap a la dreta o cap a l'esquerra o bé quan la part frontal o la part posterior són mogudes cap amunt o cap avall. Aquest efecte és conegut amb el nom de precessió.

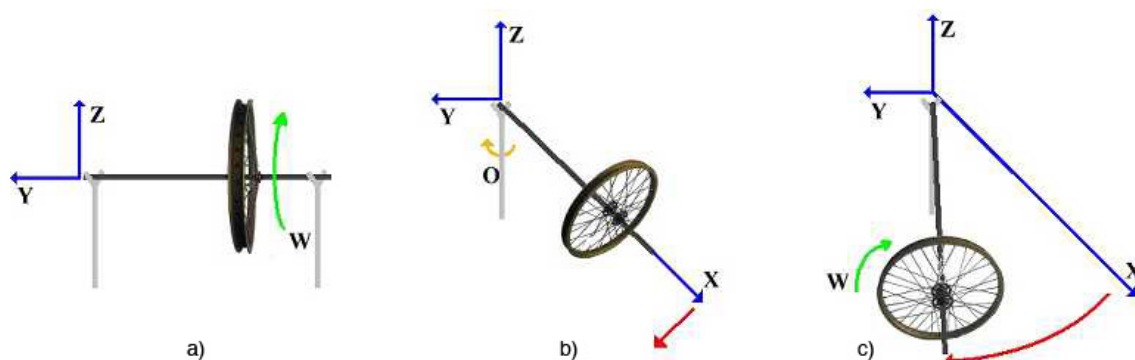


Figura 5.3: Esquema del moviment d'un giròscop

La Figura 5.3a mostra una roda de bicicleta funcionant com a giròscop. Si la roda està suportada per els dos costats de l'eix, el giròscop (la roda) simplement deixarà de girar per efecte del fregament. D'altra banda, si es suprimeix el suport dret llavors la gravetat exerceix una força atraient la part dreta de l'eix cap a baix, com es mostra a la figura 5.3b. Com s'ha explicat, la precessió giroscòpica forçarà la roda a exercir el moviment de precessió al voltant del seu eix, com és mostrat a les figures figura 5.3b i figura 5.3c per les fletxes vermelles. La direcció de la precessió depèn de la direcció de rotació dels giròscops, mostrada a les figures per les fletxes verdes. Si la roda es deixa lliure llavors continuarà rotant al voltant del seu eix (el pol).

### 5.1.2 Característiques

El sensor iGC es basa en un sofisticat conjunt de sensors que mesuren acceleració, camp magnètic i canvi d'orientació en un sistema de tres eixos ortogonals. Les dades de tots aquests sensors són processades per un microcontrolador integrat utilitzant tècniques avançades de processament de senyal per obtenir dades d'orientació en *yaw* (*heading*), *pitch* i *roll* i d'acceleració en *heave*.

Presenta un encapsulat compacte en forma d'elipsoide capaç de resistir profunditats de fins a 4000 metres. Aquest encapsulat arriba segellat i amb una verificació escrita del fabricant, fet que impedeix accedir al seu interior sense trencar-ne la garantia.

El sensor s'entrega acompanyat d'un programari que permet visualitzar *heading*, *roll*, *pitch*, nombre de girs realitzats, acceleració en *heave* i estat del dispositiu d'una manera gràfica, com si es tractés del tauler d'instruments d'una aeronau. Aquest programari s'utilitza per a comprovar el funcionament del sensor o per a tasques de calibració. Pel que fa als objectius d'aquest projecte, aquest programari no hi tindrà un paper gaire important, tot i que serà útil per a un primer contacte amb el dispositiu.

Amb la finalitat de ser útil en el major nombre d'aplicacions possible, les dades de sortida de l'iGC poden seleccionar-se per complir amb diversos estàndards de la indústria:

és compatible amb els protocols RS232, RS485 i ARCnet i també amb diversos formats de trama, tant estàndards com propietaris. Per seleccionar la sortida, cal configurar correctament una sèrie de jumpers i switches que es troben dins l'encapsulat segellat del sensor; pel que fa al cas que ens ocupa, l'iGC ha estat configurat de fàbrica per treballar en conjunt amb el sensor iFG utilitzant el protocol RS232 i no caldrà fer-hi cap modificació.

Pel que fa a les característiques elèctriques, l'iGC utilitza nominalment 24V DC a 250mA, però admet voltatges en el rang dels 12V als 26V.

A causa que l'iGC es basa en sensors que detecten el nord magnètic, és sensible a camps magnètics externs que no són produïts per la magnetosfera terrestre. Aquests camps externs poden variar en intensitat i direcció, i afecten el rendiment i la precisió de la sortida del sensor proporcionalment a la seva magnitud.

Ubicant curosament el sensor en el sistema on s'integra (vegi's el capítol 2 sobre l'Ictineu<sup>AUV</sup>), els errors provocats per la presència de petites quantitats de material ferromagnètic en l'entorn poden evitar-se, però altres efectes externs (com per exemple la presència de les parets de formigó armat de la piscina del CIRS o la proximitat dels propulsors del robot) poden produir resultats no desitjables. Com es veurà més endavant, aquest és un punt crític per aconseguir que el sensor operi correctament.

Les connexions elèctriques cap a l'exterior es fan mitjançant un connector Subconn de 8 pins (els connectors Subconn són els més utilitzats per a aplicacions submarines, ja que resisteixen grans profunditats).

L'iGC incorpora també un transductor que detecta una possible presència d'aigua dins l'encapsulat del dispositiu.

En el datasheet del sensor iGC, el fabricant proporciona les especificacions mostrades a la taula 5.1.

## 5.2 El sensor iFG (Intelligent Fibre-optic Gyro)

### 5.2.1 Principis físics

Un giròscop de fibra òptica és un dispositiu capaç de detectar rotacions mecàniques mitjançant la interferència de feixos de llum. L'aparell consisteix en una bobina de cable de fibra òptica d'uns 5000 metres de longitud. Dos feixos lumínics es fan viatjar a través de la fibra en direccions oposades. A causa de l'efecte Sagnac, el feix que viatja en sentit contrari a la rotació experimenta un camí més curt que l'altre. El desplaçament de fase resultant genera un patró d'interferències que permet conèixer la velocitat de rotació a partir de la intensitat del feix lumínic resultant.

Aquest giroscopi, conegut per les seves sigles en anglès FOG (Fibre-Optic Gyro), aporta informació molt precisa de la rotació a la qual és sotmès. A diferència del

|                                           |                                          |
|-------------------------------------------|------------------------------------------|
| <b>Rang d'orientació:</b>                 | 360°                                     |
| <b>Rang de velocitat angular:</b>         | $\pm 300^\circ/\text{s}$ (màx.)          |
| <b>Resolució del sensor:</b>              | 16 bits                                  |
| <b>Resolució de l'angle d'orientació:</b> | $<0.1^\circ$                             |
| <b>Precisió nominal:</b>                  | Millor que $1^\circ$                     |
| <b>Sortida digital:</b>                   | Sèrie RS232, RS485                       |
| <b>Freqüència de sortida:</b>             | Seleccionable fins a 20 Hz.              |
| <b>Alimentació:</b>                       | 24V DC a 100 mA. nominal                 |
| <b>Temperatura d'operació:</b>            | $-40^\circ\text{C}$ a $70^\circ\text{C}$ |
| <b>Pes a l'aire:</b>                      | 0.95 Kg.                                 |
| <b>Pes a l'aigua:</b>                     | 0.45 Kg.                                 |
| <b>Diàmetre:</b>                          | 79 mm.                                   |
| <b>Alçada:</b>                            | 121.5 mm.                                |
| <b>Profunditat màxima:</b>                | 4000 m.                                  |

Taula 5.1: Taula de característiques del sensor iGC.

giròscop tradicional, aquest no té parts mòbils ni resistència al moviment. Actualment, els giroscopis de fibra òptica són els dispositius més avançats i precisos per a la mesura de rotacions mecàniques.

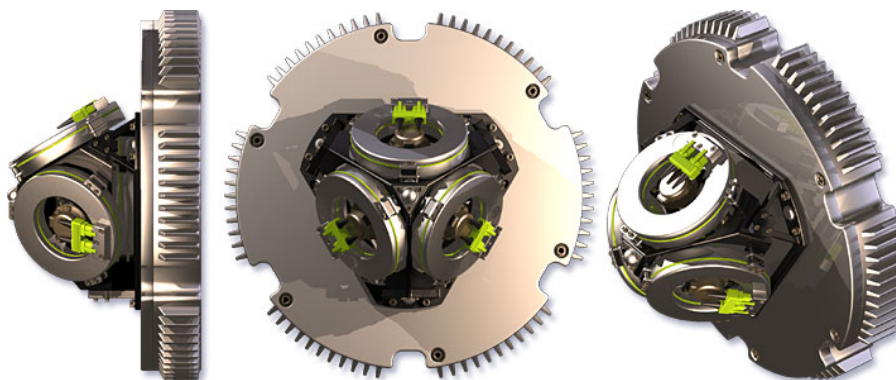


Figura 5.4: Giròscop de fibra òptica de tres eixos per a un satèl·lit orbital.

### 5.2.2 Característiques

El sensor iFG consta d'un giròscop de fibra òptica amb un encapsulat d'alumini extruït. Els giroscopis de fibra òptica constitueixen la tecnologia més avançada actualment per a la mesura de rotacions mecàniques, permetent obtenir mesures d'altíssima precisió i baixa deriva (segons el fabricant, menor a  $1^\circ$  per hora). A diferència de l'iGC, l'iFG no

està pensat per operar directament exposat a l'aigua i cal un encapsulat addicional si s'hi vol treballar en un entorn subaquàtic.

Com que l'iFG utilitza la interferència lumínica produïda entre dos feixos de llum per mesurar la rotació mecànica, aquesta mesura és relativa a la seva posició inicial. Si es vol obtenir una mesura relativa a un punt de referència com per exemple el nord magnètic de la Terra, cal utilitzar un dispositiu capaç de mesurar la orientació relativa a aquest punt. Per tant, l'iFG està pensat per operar com a parella de l'IGC o d'una altra unitat inercial, inicialitzant-se a paritr d'una mesura d'aquesta última.

El sensor presenta dos connectors DE-9 (els típicament utilitzats en els PC per als ports sèrie), anomenats port 1 (a l'esquerra) i port 2 (a la dreta). Al port 1 s'hi connecta l'alimentació elèctrica i la sortida de la unitat inercial iGC com a entrades, i opcionalment s'hi pot connectar també un port sèrie de PC expressament destinat a utilitzar-se amb el programari proporcionat pel fabricant. Al port 2 és on pot connectar-se el port sèrie d'un PC o d'un sistema de control per a processar les cadenes de text que envia el sensor amb les mesures de tot el sistema.

El format de la sortida del sensor pot configurar-se mitjançant dos dispositius: un és un selector de 10 posicions que es troba al frontal de l'aparell; l'altre consisteix en un conjunt de quatre interruptors que es troben a l'interior (cal desmuntar el frontal per accedir-hi). Els protocols suportats són RS232 i RS485, a baud rates de 4800, 9600 i 19200 bauds. Al frontal del sensor també hi ha sis díodes LED a mode d'indicadors d'estat del sistema.

Una característica remarcable de l'iFG és que no ve acompanyat de cap manual que en detalli el funcionament: únicament es pot disposar de les referències complementàries que s'hi fan al manual del seu company iGC i d'unes taules que indiquen els modes de sortida que ofereix l'iFG. A causa d'això, per estudiar-ne el funcionament va ser necessari un intens contacte via correu electrònic amb els enginyers de Tritech, el fabricant del dispositiu. A més, van descobrir-se diversos errors a l'escassa documentació proporcionada, com es detalla més endavant.

En el datasheet del sensor iFG, el fabricant proporciona les especificacions mostrades a la taula 5.2.

## 5.3 Els sensors iGC i iFG com a conjunt sensorial

### 5.3.1 Procés de sincronització

Com ja s'ha comentat, la unitat de referència inercial iGC i el giròscop de fibra òptica iFG estan pensats per funcionar sincronitzadament. A continuació es detalla el procés que es duu a terme per sincronitzar els dispositius:

1. Quan s'engega el sistema (en donar pas al corrent elèctric que l'alimenta), la unitat

|                                           |                                                  |
|-------------------------------------------|--------------------------------------------------|
| <b>Rang d'orientació:</b>                 | 360° en un sol eix (eix de /emphyaw)             |
| <b>Rang de velocitat angular:</b>         | $\pm 375^\circ/\text{s}$                         |
| <b>Resolució de l'angle d'orientació:</b> | 0.01°                                            |
| <b>Precisió nominal:</b>                  | 0.1° (depenent del protocol de sortida escollit) |
| <b>Derival:</b>                           | Nominalment $\pm 5^\circ$ per hora               |
| <b>Temps d'inicialització:</b>            | 5 segons                                         |
| <b>Sortida digital:</b>                   | Sèrie RS232, RS485 (dos ports)                   |
| <b>Alimentació:</b>                       | 10 a 30V DC, 5W màxim                            |
| <b>Temperatura d'operació:</b>            | -40° C a 70° C                                   |
| <b>Pes:</b>                               | 0.5 Kg.                                          |
| <b>Mides:</b>                             | 110 × 45 × 90 mm. (excloent connectors)          |

Taula 5.2: Taula de característiques del sensor iFG.

- inercial iGC troba la orientació relativa al nord del dispositiu en aquell moment.
- Immediatament després, aquesta dada és enviada al giròscop de fibra òptica iFG.
- A continuació el giròscop de fibra òptica es sincronitza amb aquesta orientació, procés que tarda uns 30-40 segons.
- Un cop l'iFG s'ha sincronitzat, aquest esdevé la unitat de referència per a la mesura de la orientació.

Evidentment, aquest procés de sincronització només es duu a terme pel que fa a la orientació en yaw del sistema. La unitat inercial iGC segueix essent el punt de referència per a les mesures de rotació en roll i pitch i d'acceleració en heave.

### 5.3.2 Interconnexió dels dispositius

L'iGC i l'iFG poden connectar-se en un ampli ventall de configuracions que permeten la compatibilitat amb diversos estàndards. Però pel que fa als nostres interessos (obtenir un sistema que informi de la orientació en yaw amb la major precisió possible a través del protocol RS232), la connexió dels dispositius seguirà el següent esquema:

Així, la sortida de l'iGC es connecta al port 1 de l'iFG, on l'iGC transmet les dades de heading, roll pitch i acceleració en heave en un format propietari. A través d'aquest mateix port també rep alimentació tot el sistema. Mitjançant el port 2, l'iFG transmet la informació amb el heading corregit amb la lectura del giròscop de fibra òptica i la resta de dades rebudes des de l'iGC mitjançant un protocol RS232 i un format de trama estàndard, escollit per l'usuari com es detalla en el següent apartat. És en aquest port 2 on es connecta un enllaç de línia sèrie RS232 que es connecta a un PC per tal de processar les dades.

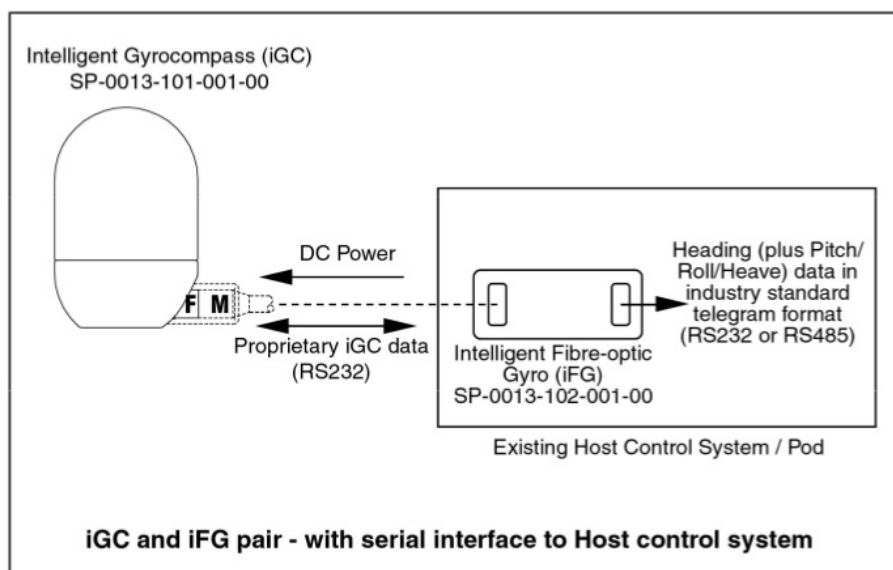


Figura 5.5: Esquema de connexió de parell de sensors iGC i iFG

### 5.3.3 Modes d'operació i formats de sortida

En el sistema de sensors format per l'iGC i l'iFG existeixen tres enllaços de comunicació digital. D'aquests tres, l'únic que té interès per a l'usuari final és el que uneix el port 2 de l'iFG amb el PC del sistema de control. Dels altres dos enllaços, un és utilitzat pels dos sensors per transmetre's dades entre sí i l'altre pot connectar-se opcionalment a un PC per prendre dades del sistema de sensors fent servir el programari que proporciona el fabricant (per exemple per realitzar tasques de comprovació del sistema o de calibració).

Així, doncs, és interessant estudiar en quin format es transmeten les dades des del port 2 de l'iFG fins al PC on es connecta per tal de poder rebre-les i processar-les. Segons les taules proporcionades per Tritech, pel port 2 de l'iFG es poden transmetre dades tant en format RS232 com en format RS485, a diferents velocitats de transmissió o baud rates. Tant per un protocol com per l'altre, es poden escollir fins a 16 formats de trama diferents consistents en cadenes de text ASCII. No totes les cadenes de text transmeten la mateixa informació, i caldrà tenir en compte quina d'elles és la que més s'ajusta a les necessitats de l'Ictineu<sup>AUV</sup> a l'hora d'integrar els sensors.

La selecció del protocol de transmissió, la velocitat o baud rate i el format de trama es fa mitjançant dos selectors presents al sensor iFG. Un d'ells, situat a la part frontal del sensor i anomenat BCD Mode Switch, permet seleccionar el protocol de transmissió (RS232 o bé RS485) i la velocitat de transmissió (19200, 9600 o 4800 baud). L'altre selector s'anomena Submode Switch, i permet escollir el format de trama desitjat. Aquest segon selector es troba dins l'encapsulat del sensor i per accedir-hi cal obrir-ne la tapa frontal extraient els visos que la subjecten, tal com es mostra a la figura ??.

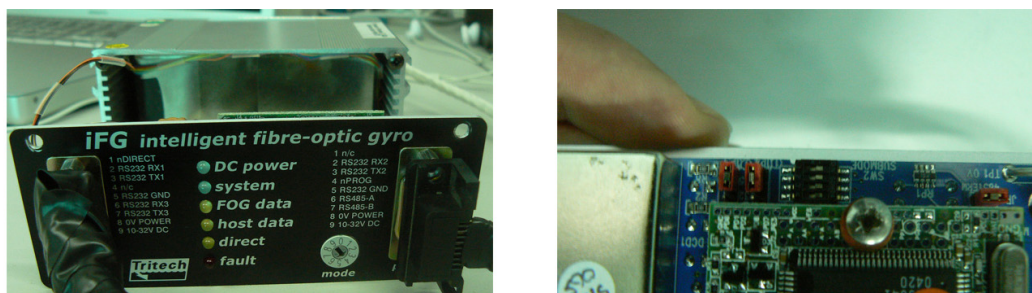


Figura 5.6: Selectors de mode de l'iFG. A la dreta pot veure's el BCD Mode Switch, al frontal del dispositiu. A l'esquerra pot veure's el selector Submode switch.

### 5.3.4 Calibració del sistema

Les distorsions magnètiques que provoquen desviacions constants en les mesures de l'iGC poden compensar-se mitjançant un procés de calibració. Per a dur a terme aquest procés, cal un equipament addicional que mesuri les mateixes magnituds que el sensor. Aquest equipament ha de tenir una precisió major que els valors que volen calibrar-se.

Aquest procés consisteix en connectar un PC directament a l'iGC (configurat en un mode concret, fet que implica obrir l'encapsulat segellat del sensor) i rotant el sensor a un valor de heading conegut. A continuació es prenen els valors de tots dos dispositius, que haurien de ser iguals en absència de distorsions, i s'introdueix la diferència al software de calibració proporcionat per Tritech. Aquest mateix procés es repeteix per a diversos punts dels 360° de rotació. Llavors, a partir d'aquests punts el microcontrolador de l'iGC calcula una corba de compensació extrapolant entre els punts i la desa a memòria. Com més punts es prenen, més acurada resulta la corba de compensació.

Tot i que la teoria en la que es basa aquest procediment és àmpliament coneguda, el fabricant no proporciona les instruccions per a calibrar el sensor. L'únic que s'indica és que pot contactar-se amb el fabricant per a més informació. D'aquí es dedueix que aquest procediment no és trivial.

## 5.4 Experiments previs a la integració dels sensors a l'Ictineu<sup>AUV</sup>

Un cop estudiades les característiques i el funcionament dels sensors iGC i iFG, s'han dut a terme un conjunt d'experiments amb la finalitat de comprovar el rendiment i la precisió dels sensors i de comprendre les dades enviades cap al PC on han de ser processades.

Per a poder fer això, cal realitzar una instal·lació elèctrica (temporal) dels dispositius: cal connectar la parella de sensors entre sí i connectar una font d'alimentació al sistema per proporcionar-li el corrent elèctric necessari. Aquesta instal·lació s'ha realitzat seguint



l'esquema mostrat a la figura 5.5, i un cop finalitzada presenta l'aspecte mostrat a la figura 5.7.

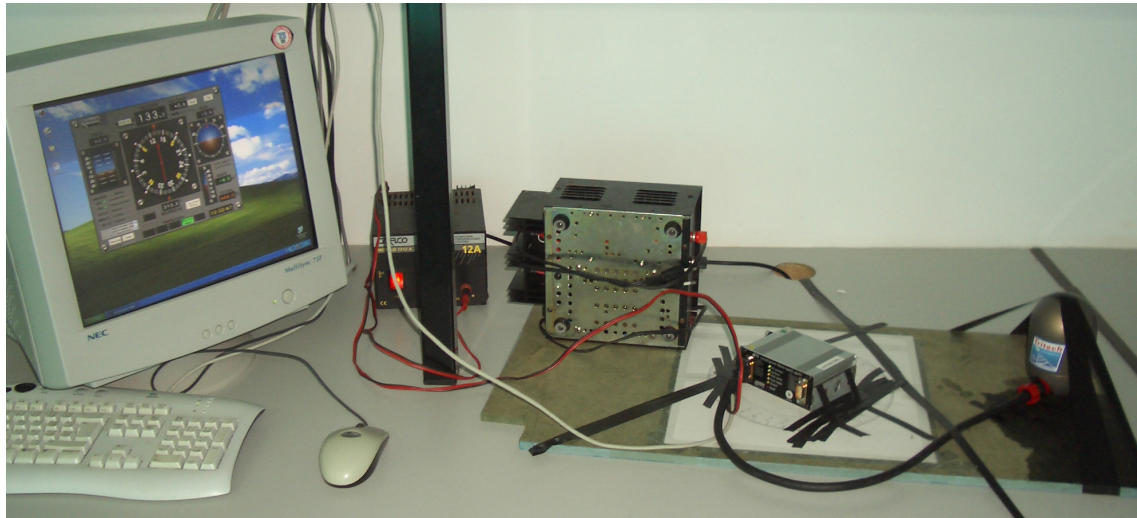


Figura 5.7: Banc de proves per als sensors iGC i iFG

Un cop feta la instal·lació elèctrica, el sistema de sensors es connecta, mitjançant un cable sèrie RS232, a un PC amb el sistema operatiu Windows i el programari proporcionat per Trittech instal·lat. S'utilitzarà aquest programari per a prendre les dades necessàries. També es farà ús d'un terminal sèrie (HyperTerminal) per llegir la sortida ASCII transmesa pels sensors.

#### 5.4.1 Anàlisi de la sortida del sistema de sensors

El primer experiment consisteix en comprovar, configurant els sensors amb tots els formats de trama possibles, que la sortida s'adiu amb el que especifica el fabricant a la documentació proporcionada. Per fer aquesta comprovació, es configurarà el sensor perquè transmeti utilitzant el protocol RS232 i el baud rate més alt possible, 19200 baud. S'ha seleccionat la major velocitat de transmissió perquè també és la que proporciona un ratio d'actualització de les dades més elevat (20Hz). Aquesta velocitat de transmissió és suportada sense problemes per el PC, permetent treballar en temps real (el ratio d'actualització mínim perquè una tasca es consideri que treballa en temps real és 10Hz). Aquesta configuració implica posar el BCD Mode Switch (el selector de la part frontal del sensor iFG) a la posició 1.

Per provar tots els formats de trama disponibles, caldrà moure el selector Submode Switch que es troba dins l'encapsulat de l'iFG. Aquest selector consisteix en un conjunt de 4 interruptors binaris, que es poden disposar en 16 combinacions diferents (de la 0 a la 15). Escollint l'identificador de trama indicat a les taules proporcionades pel fabricant, es selecciona la trama desitjada. El format de trama, però, no depèn només

| BCD | Sub | Sortida esperada                  | Coincideix |
|-----|-----|-----------------------------------|------------|
| 1   | 0   | 4 bytes en seqüència              | SÍ         |
| 1   | 1   | 4 bytes en seqüència              | SÍ         |
| 1   | 2   | \$HHH.H,N,SS                      | SÍ         |
| 1   | 3   | \$HHH.H,N,SS                      | SÍ         |
| 1   | 4   | \$HHH.H,N,SS,+LL,QQ,+RR.R,+PP.P   | SÍ         |
| 1   | 5   | \$HHH.H,N,SS,+LL,QQ,+RR.R,+PP.P   | SÍ         |
| 1   | 6   | HHHH                              | SÍ         |
| 1   | 7   | \$HEHDT,hhh.h,T*kk                | SÍ         |
| 1   | 8   | \$HCHDM,hhh.h,M*kk                | SÍ         |
| 1   | 9   | HhhhhPsppppRrrrrf                 | SÍ         |
| 1   | 10  | HhhhhPsppppRrrrrf                 | SÍ         |
| 1   | 11  | 18 bytes en seqüència             | NO         |
| 1   | 12  | Seqüència de tres trames          | SÍ         |
| 1   | 13  | Seqüència de tres trames          | SÍ         |
| 1   | 14  | \$PTVG,abbbbPaccccR,ddd.dT*kk     | SÍ         |
| 1   | 15  | HhhhhPsppppRrrrrZsaaaafw          | NO         |
| 4   | 15  | HhhhhCccccPsppppRrrrrZsaaaaMmmmfw | NO         |

Taula 5.3: Comprovació de la sortida del sistema de sensors.

del Submode Switch. Existeixen dues taules de formats de trama a la documentació del sensor, la taula 1 i la taula 2. Per a una velocitat de transmissió de 19200 baud, si la posició del selector BCD Mode Switch és 1, el format de trama es selecciona de la taula 1 d'acord amb la posició del Submode Switch. Si la posició del selector BCD Mode Switch és 4, el format de trama es selecciona de la taula 2.

Val a dir, però, que la taula 1 disposa de 16 formats de trama (identificats del 0 al 15) i en canvi la taula 2 només disposa d'un sol format per escollir (identificat amb el valor 15). Possiblement el fabricant pretén incloure nous formats a la taula 2 en futures actualitzacions del firmware dels sensors, però de moment les configuracions possibles són aquestes.

Així, el procediment experimental consisteix en configurar el sensor en un mode concret i comprovar que la sortida es correspon amb el que s'especifica a la documentació. Per tant, caldrà provar les 16 combinacions de trama possibles:

Com es mostra a la taula 5.3, hi ha tres formats de sortida que no es corresponen amb la documentació. Aquests són els següents:

- *BCD Mode=1, Submode=11*. La sortida real mostra un missatge dient que el telegrama encara no està implementat al *firmware* del dispositiu.

- *BCD Mode=1, Submode=15*. La sortida real difereix en un caràcter.
- *BCD Mode=4, Submode=15*. La sortida real difereix en diversos caràcters al final de la trama.

Per a solventar això, es va contactar amb el fabricant i aquest va determinar que el problema venia donat perquè la versió de la documentació proporcionada no coincidia amb el *firmware* del sensor. Trittech va enviar nova documentació actualitzada i el problema va quedar resolt.

### 5.4.2 Anàlisi de l'efecte provocat pels camps magnètics aliens a la magnetosfera terrestre

Per comprovar quins efectes tenen els camps magnètics provocats per elements aliens al camp magnètic terrestre sobre les mesures preses pel sensor iGC (l'únic del parell que és sensible a aquestes distorsions) s'ha realitzat el següent experiment. Consisteix senzillament en apropar diferents cossos metàl·lics al sensor i observar com varia la seva sortida. El procediment s'ha aplicat emprant tres cossos diferents: un tornavís sol, un conjunt de tres tornavisos i un pic. La figura ?? mostra els objectes utilitzats.



Figura 5.8: Cossos metàl·lics utilitzats per a comprovar els efectes de la distorsió magnètica. D'esquerra a dreta: un tornavís, un conjunt de tres tornavisos i un pic.

El sensor s'ha mantingut fixat a una estructura de manera que la seva orientació no canviï quan no hi ha presència de distorsions magnètiques. Cada objecte s'ha apropat a 20, 10 i 3 cm. de l'encapsulat del sensor i s'ha mesurat la desviació obtinguda en la mesura. La taula 5.4 mostra els resultats obtinguts.

Com podia preveure's, la desviació augmenta proporcionalment a la mida del cos metàl·lic i a la proximitat d'aquest.

### 5.4.3 Anàlisi de la deriva del sistema

Pel que fa al giròscop de fibra òptica, l'element principal del sistema de sensors ja que és el que permetrà disposar d'una mesura precisa i lliure de distorsió magnètica del heading

| Objecte         | Desviació a 20 cm | Desviació a 10 cm | Desviació a 3 cm |
|-----------------|-------------------|-------------------|------------------|
| Tornavís        | 2°                | 6°                | 9°               |
| Tres tornavisos | 2°                | 8°                | 12°              |
| Pic             | 5°                | 11°               | 21°              |

Taula 5.4: Resultats obtinguts amb l'experiment de comprovació de distorsió magnètica.

| Experiment | Temps   | Desviació absoluta | Deriva       |
|------------|---------|--------------------|--------------|
| Dinàmic    | 5401 s. | 0.2°               | 0.13 °/hora  |
| Estàtic    | 4801 s. | 22.4°              | 16.79 °/hora |

Taula 5.5: Resultats obtinguts amb l'experiment de comprovació de la deriva.

del robot, es comprovarà com deriva la mesura proporcionada pel sensor en l'interval de temps d'una hora.

Aquest experiment es durà a terme en dues situacions diferents. Primer, el sistema de sensors restarà en estat estàtic (sense que els sensors canviïn la seva posició o orientació relativa) i es mesurarà com ha variat el valor indicat pel sensor en una hora. La figura 5.9 mostra l'evolució de les mesures de heading del sensor al llarg d'una hora.

En el segon cas, s'iniciarà l'experiment amb els sensors situats en posició absoluta i relativa coneguda. A continuació, durant una hora, s'aplicaràn moviments arbitraris al sistema de sensors. Finalment, transcorregut aquest temps es tornarà a situar els sensors en la posició inicial i es comprovarà com ha variat la mesura del heading. És molt important que els moviments s'apliquin amb els dos sensors conservant la seva orientació relativa entre sí. La figura 5.10 mostra l'evolució de les mesures de heading al llarg d'una hora i la diferència entre el valor inicial i el valor final

A la taula 5.5 es mostra el temps d'execució, els valors inicials i finals de heading i la diferència entre aquests (la deriva) per a cada un dels dos experiments. Pot observar-se que amb els sensors en moviment la deriva obtinguda és de 16.79 °/hora, notablement superior als 5 °/hora màxims que estipula el fabricant. Per això, s'està en contacte amb els enginyers de Tritech per a poder realitzar una calibració dels sensors que permeti treure'n el màxim rendiment.

#### 5.4.4 Conclusions de l'estudi dels sensors

A continuació s'expliquen les conclusions extretes dels tres experiments realitzats amb els sensors iGC i iFG.

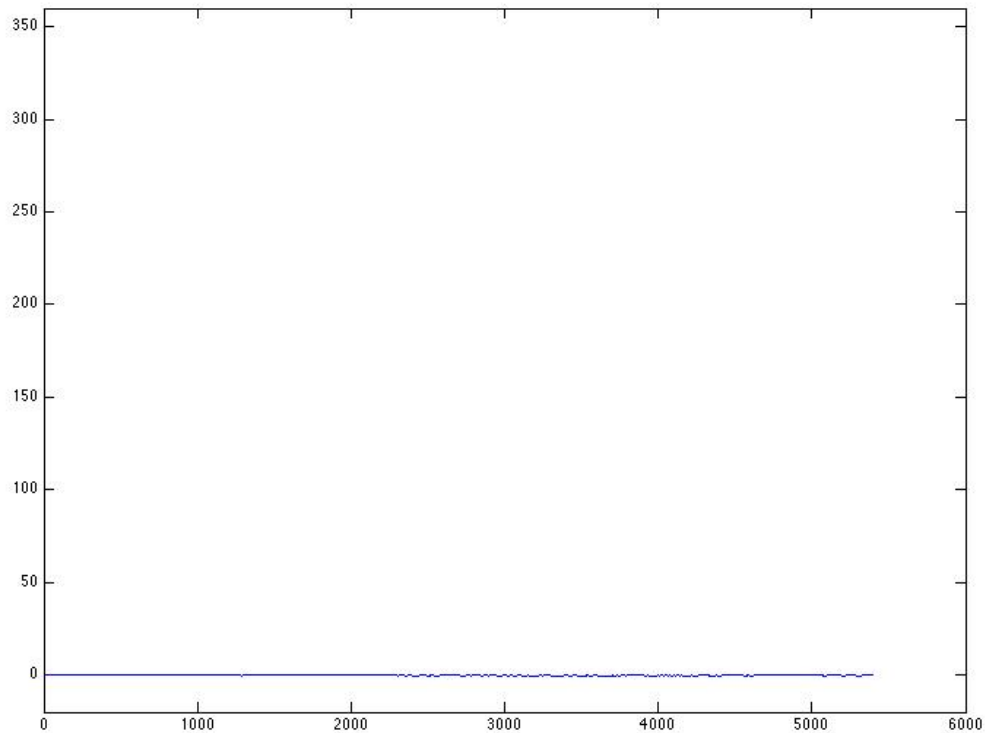


Figura 5.9: Evolució de la mesura d'orientació al llarg del temps amb els sensors en estat estàtic.

### Conclusions de l'anàlisi dels formats de trama

Els tres formats de trama no documentats descoberts han permès solventar el contratemps que representaria aquesta incoherència a l'hora de desenvolupar un programari que processés les dades dels sensors.

Analitzant els diferents formats de sortida, s'ha decidit que el més útil és l'obtingut amb l'interruptor BCD Mode en la posició 4 i l'interruptor Submode a la posició 15, ja que aquesta trama és la única que informa de totes les dades que el sistema és capaç de proporcionar. Per tant, serà aquest el format que s'utilitzarà quan s'instal·li el sensor al robot, ja que llavors no serà possible canviar fàcilment el mode de sortida.

### Conclusions de l'anàlisi de l'efecte de la distorsió magnètica

La sortida del sensor iGC es veu seriosament afectada per la presència de cossos metàl·lics al seu entorn, podent donar un error de fins a  $20^\circ$  en la mesura de l'angle d'orientació.

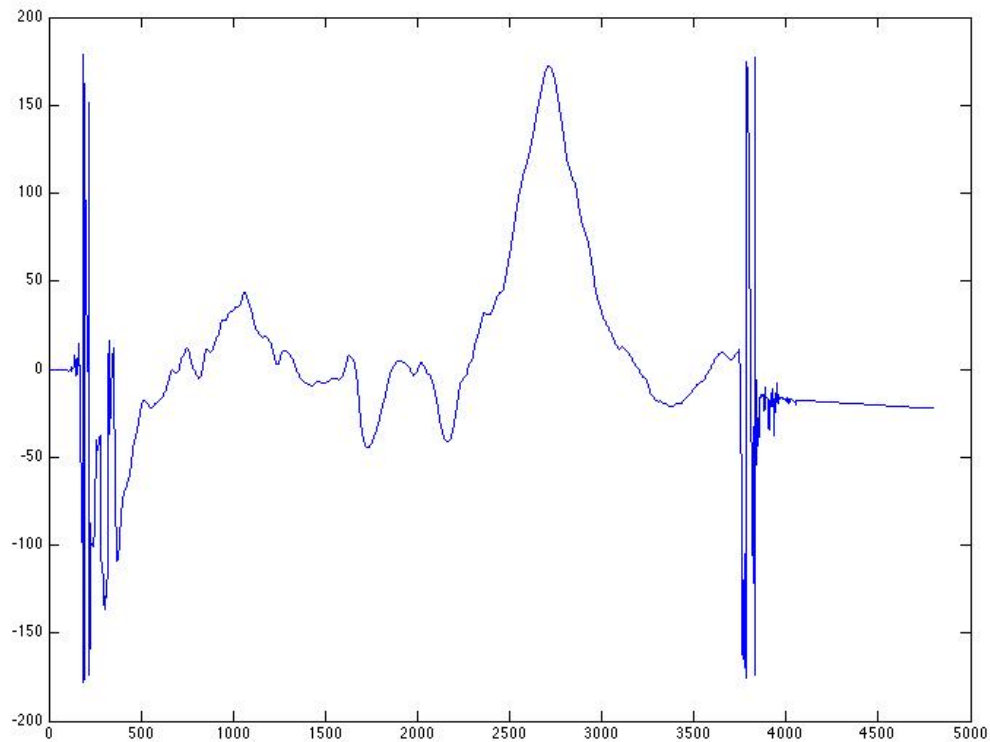


Figura 5.10: Evolució de la mesura d'orientació al llarg del temps amb els sensors en moviment.

Evidentment, aquest error pot ser encara major si s'exposa el sensor a la presència d'un cos metàl·lic més massiu i voluminós. Així, hom pot fer-se una idea del que succeirà dins la piscina quan el sensor s'acosti a una de les parets de formigó armat.

### Conclusions de l'anàlisi de la deriva del sistema

Pot observar-se que quan el sistema de sensors està en moviment, i així ho estarà quan el robot executi una missió, el sistema té una deriva de  $16.79^\circ/\text{hora}$ . S'espera que Trittech proporcioni la informació necessària per a realitzar un procés de calibració que permeti optimitzar aquest valor de deriva. Tot i això, a l'hora de finalitzar aquest projecte el procediment encara no s'ha pogut realitzar.

# Capítol 6

## Integració dels sensors iGC i iFG a l'Ictineu<sup>AUV</sup>

Un cop estudiat el funcionament dels nous sensors, el següent pas a fer és integrar-los al robot Ictineu<sup>AUV</sup>. En aquest capítol s'explica com s'ha dut a terme la instal·lació mecànica i elèctrica dels dispositius, així com el desenvolupament del programari necessari per afegir els nous sensors a l'arquitectura de control. També es descriuen els experiments duts a terme per comprovar la precisió i la deriva dels sensors un cop integrats al robot.

### 6.1 Integració dels sensors a l'estructura del robot

Per poder fer un lloc als nous sensors, primer cal tenir clar quines connexions elèctriques s'hauran d'afegir a les existents per tal d'alimentar els dispositius i transmetre'n les dades. Un cop es té clara la disposició dels cables, es procedeix a fer la instal·lació mecànica dels dispositius mirant de complir al màxim els requeriments especificats pel fabricant, explicats més endavant.

#### 6.1.1 Instal·lació elèctrica

La connexió dels sensors segueix bàsicament l'esquema presentat al punt 5.3.2. A la figura 6.2 es mostra l'esquema detallat, pin a pin, del cablejat necessari per connectar els sensors entre ells i al PC de control del robot.

Cal notar que del port 1 de l'iFG surt una línia sèrie RS232 a través de la qual es transmeten les dades d'orientació, acceleració i estat del sistema en un format propietari. Aquest cable pot ser utilitzat per si es vol fer servir el programari inclòs amb els sensors amb la finalitat de realitzar tasques de calibració o de comprovació de sortida. El cable

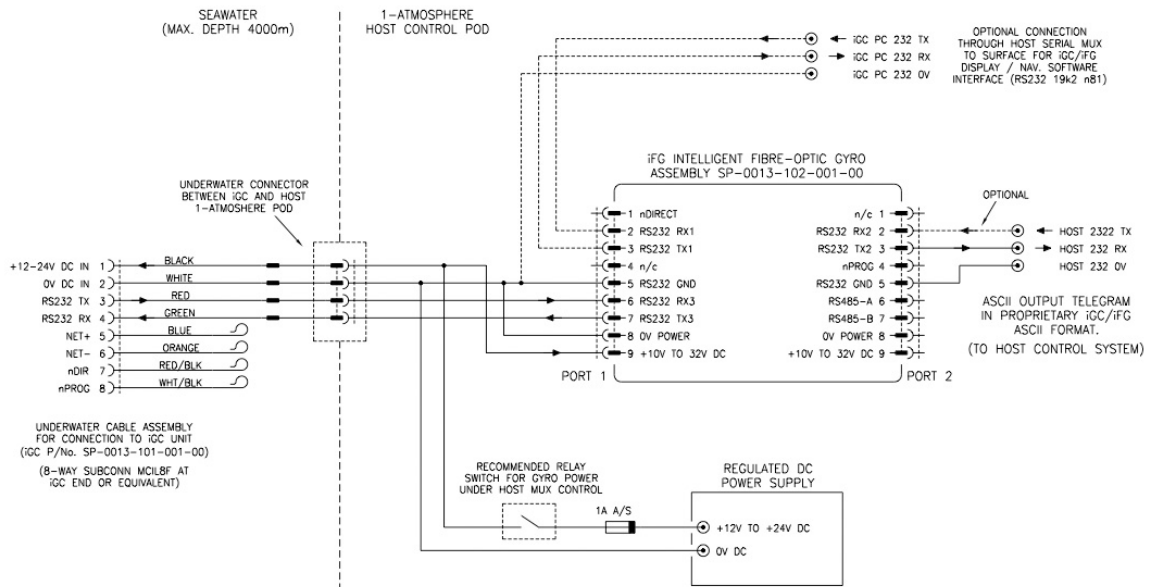


Figura 6.1: Esquema elèctric detallat del connexionat dels sensors.

no s'utilitzarà normalment, però la connexió s'ha dut a terme de totes maneres ja que pot ser útil en certes circumstàncies.

### 6.1.2 Instal·lació mecànica

Un cop determinat el cablejat necessari per a instal·lar els nous sensors, s'ha procedit a realitzar-ne la instal·lació mecànica. Cada sensor s'ha integrat al robot d'acord als seus requeriments.

#### Instal·lació mecànica de l'IGC

Pel que fa l'IGC, dispositiu ja pensat per a operar directament en un entorn subaquàtic, només ha calgut afegir una estructura subjectada entre els dos cilindres del robot a la qual s'ha unit el sensor mitjançant unions cargolades.

Degut al poc espai disponible i a les exigències del cablejat, no ha estat possible ubicar els eixos del sensor anàlogament als de l'Ictineu<sup>AUV</sup>, havent hagut d'aplicar una rotació de 90° al sensor (respecte el sistema de coordenades del robot). Així, doncs, caldrà aplicar la següent matriu de canvi de sistema de coordenades a les mesures rebudes des del sensor:



$$\begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Aplicar aquesta matriu significa, en la pràctica, que el nou roll és el pitch del sistema de coordenades original, i que el nou pitch és el roll original canviat de signe. Pel que fa al heading només cal restar  $90^\circ$  al valor proporcionat pel sensor per a obtenir el valor de heading del robot.

### Instal·lació mecànica de l'iFG

La instal·lació de l'iFG ha resultat força més complicada, sobretot pel fet que l'encapsulat d'aquest sensor no està dissenyat per a permetre operar sota l'aigua. Per tant, ha calgut construir un cilindre addicional per allotjar el sensor, que s'ha situat prop del centre de gravetat del robot, damunt del DVL. El sensor s'ha instal·lat amb els eixos solidaris als de l'iGC (els sistemes de coordenades dels dos sensors no estan rotats l'un respecte l'altre). Per tant, també cal aplicar l'anterior matriu de rotació als valors proporcionats per l'iFG (procediment que es tradueix, igualment, en restar  $90^\circ$  al valor de heading obtingut del sensor).



Figura 6.2: Els sensors giroscòpics instal·lats al robot.

## 6.2 Integració dels sensors a l'arquitectura de control O2CA2

Ja amb els sensors instal·lats al robot i amb les connexions fetes, s'ha procedit a desenvolupar el programari necessari per integrar el parell de sensors al l'arquitectura de control de l'Ictineu<sup>AUV</sup>. Per a integrar un nou sensor a l'arquitectura O2CA2, existeix un protocol no escrit que consisteix en els següents dos passos:

1. Es desenvolupa una classe en C++ que, mitjançant els seus mètodes, permeti interactuar amb el sensor rebent-ne dades i enviant-n'hi si és necessari. Un cop s'ha construït aquesta classe, es crea un programa principal que utilitza els mètodes de la classe per mostrar la informació enviada pel sensor a la pantalla del PC on s'executa el programa.
2. Un cop es disposa d'aquest programari, es construeix una interfície CORBA amb els mateixos mètodes i estructures de dades que la classe anteriorment desenvolupada. Un cop compilada la interfície, s'omple l'esquelet de l'objecte CORBA generat reciclant el codi de la classe original.

Seguint aquest procés s'evita haver de treballar amb les particularitats de CORBA i s'assegura un correcte funcionament del codi que interactua amb el sensor. Així, doncs, per integrar el conjunt de sensors que ens ocupa es realitzarà aquest mateix procediment.

### 6.2.1 Desenvolupament de l'aplicació en C++

#### Disseny

L'aplicació en C++ s'ha dissenyat per a poder funcionar amb tots els formats de sortida de l'IFG disponibles, tot i que com ja s'ha comentat al punt 5.4.4 els sensors només funcionaran en un sol mode un cop integrats. El disseny es basa en una classe que té la següent signatura:

```
class CFOGCL
{
 public:
 CFOGCL(std::string serialPortPath, int mode, int submode);

 SFOGCLInit Init();

 void Iteration();
 void Show();

 std::string LogData();

 void SyncFogHeading(double syncValue);
```

```

double GetCorrectedFogHeading ();
double GetRawFogHeading ();
double GetMagneticHeading ();
double GetPitch ();
double GetRoll ();
double GetHeave ();

double GetHeadingVelocity ();
double GetPitchVelocity ();
double GetRollVelocity ();

double GetFogHeadingOffset ();

SFOGCL_T1S0 ReadingT1S0 ();
SFOGCL_T1S1 ReadingT1S1 ();
SFOGCL_T1S2 ReadingT1S2 ();
SFOGCL_T1S3 ReadingT1S3 ();
SFOGCL_T1S4 ReadingT1S4 ();
SFOGCL_T1S5 ReadingT1S5 ();
SFOGCL_T1S6 ReadingT1S6 ();
SFOGCL_T1S7 ReadingT1S7 ();
SFOGCL_T1S8 ReadingT1S8 ();
SFOGCL_T1S9 ReadingT1S9 ();
SFOGCL_T1S10 ReadingT1S10 ();
SFOGCL_T1S11 ReadingT1S11 ();
SFOGCL_T1S12 ReadingT1S12 ();
SFOGCL_T1S13 ReadingT1S13 ();
SFOGCL_T1S14 ReadingT1S14 ();
SFOGCL_T1S15 ReadingT1S15 ();
SFOGCL_T2S15 ReadingT2S15 ();
}

```

Aquesta classe s'ha dissenyat ja d'entrada per a poder-la utilitzar en el desenvolupament de l'objecte CORBA per a l'arquitectura O<sup>2</sup>CA<sup>2</sup>. Inclou una operació que retorna una cadena de text formatejada d'acord als convenis del laboratori i on s'inclouen les dades obtingudes a cada lectura dels sensors a fi de desar aquesta informació en un fitxer de log. També inclou una característica que permet reinicialitzar el valor de heading obtingut dels sensors via *software*, afegint un valor de compensació (*offset*) a les mesures. Això permetrà poder sincronitzar el sensor iFG amb qualsevol altre sensor del robot capaç de proporcionar la orientació relativa al nord i no només amb l'iGC.

Anem a comentar cada un dels mètodes de la classe:

- *CFOGCL(std::string serialPortPath, int mode, int submode)*: constructor a partir de la ruta al fitxer de dispositiu del port sèrie del sensor, del valor de l'interruptor BCD Mode Switch i de l'interruptor Submode Switch.
- *Init()*: retorna una estructura que conté els valors d'inicialització especificats al constructor. Aquest mètode és present a tots els objectes de l'arquitectura, com

es veurà més endavant, i s'ha inclòs en aquesta classe C++ per a fer més fàcil la portabilitat a CORBA.

- *Iteration()*: S'encarrega de llegir les trames enviades pel sensor al port sèrie i d'interpretar-les segons el mode de funcionament seleccionat. Cal executar aquest mètode cada vegada que es vol actualitzar les mesures; típicament s'inclourà dins el cos d'un *loop* incondicional.
- *Show()*: Mostra per pantalla les darreres mesures preses.
- *LogData()*: Retorna una cadena de text formatejada d'acord al conveni utilitzat a l'arquitectura O<sup>2</sup>CA<sup>2</sup> per a generar fitxers de log. Aquesta cadena contindrà les darreres mesures preses.
- *SyncFogHeading(double syncValue)*: Aplica un offset a la mesura de heading presa del sensor. En la pràctica, aquest mètode sincronitza el heading proporcionat per l'IFG amb un valor que pot haver prè un altre sensor.
- *GetCorrectedFogHeading()*: Retorna el valor llegit del sensor amb la compensació aplicada mitjançant *SyncFogHeading(double syncValue)*.
- *GetRawFogHeading()*: Retorna el valor de heading sense aplicar-hi cap offset.
- *GetMagneticHeading()*: Retorna el valor de heading proporcionat per la unitat inercial iGC (no per el giròscop de fibra òptica).
- *GetPitch()*: Retorna el valor de l'angle d'orientació en pitch.
- *GetRoll()*: Retorna el valor de l'angle d'orientació en roll.
- *GetPitch()*: Retorna el valor de l'acceleració linial en heave.
- *GetHeadingVelocity()*: Retorna la derivada respecte el temps de l'angle d'orientació en yaw. Aquesta mesura serà útil per al sistema de navegació del robot, que es basa en velocitats.
- *GetPitchVelocity()*: Retorna la derivada respecte el temps de l'angle d'orientació en pitch.
- *GetRollVelocity()*: Retorna la derivada respecte el temps de l'angle d'orientació en roll.
- *GetFogHeadingOffset()*: Retorna el valor del l'offset aplicat a la mesura de heading.
- *ReadingTxSy()*: Per a  $1 \leq x \leq 2$  i per a  $0 \leq y \leq 15$ , retorna una estructura C++ corresponent al mode de funcionament seleccionat amb les dades obtingudes de la trama rebuda.

## Implementació

La classe s'ha implementat en C++ en una plataforma GNU/Linux, a fi d'assimilar al màxim l'entorn de desenvolupament amb l'entorn d'execució (el robot). Per a manejar els accessos al port sèrie s'ha utilitzat la biblioteca de lliure distribució libserial, que facilita l'accés al dispositiu proporcionant els mètodes típics que pot trobar-se en qualsevol stream de la biblioteca estàndard de C++.

### 6.2.2 Desenvolupament de l'objecte CORBA per a l'arquitectura O2CA2

#### Disseny

Per al disseny de l'objecte corba caldrà tenir molt en compte les característiques de disseny de l'arquitectura O2CA2. Per això, cal estudiar com ha de ser un objecte de l'arquitectura. A continuació s'explica el conveni a seguir per tal que el propi disseny i la implementació siguin el màxim d'homogenis possible:

- Tots els tipus enumerats comencen per E.
- Tots els noms de les estructures comencen per S.
- Tots els noms de les classes comencen per C.
- Tots els paràmetres d'inicialització dels diferents objectes es passen al seu constructor dins una estructura del tipus `SnomInit`.
- L'estructura de tipus `SnomInit`, a més de contenir els paràmetres d'inicialització (no modificables), conté l'estructura de configuració del tipus `SnomConfiguration` (els paràmetres de la qual sí que es poden modificar).
- Tot objecte que tingui paràmetres d'inicialització té un atribut anomenat `Init`, només de lectura, que retorna els valors assignats a la seva estructura `SnomInit`.
- Tot objecte que tingui paràmetres de configuració té un atribut anomenat `Configuration`, de lectura/escriptura.
- Tot objecte sensor té un mètode anomenat `Reading()` que retorna les mesures preses pel sensor en un instant de temps determinat dins una estructura del tipus `SnomSensor`.

Pel que fa a la integració del nou objecte a l'arquitectura, cal tenir en compte la jerarquia de classes present en aquesta. Per començar, tot objecte de l'arquitectura (excepte els objectes destinats exclusivament a la tasca de desar els logs a disc, on treballar en

temps real no és important) hereda de la interfície `CRTPeriodicThread`. Aquesta interfície proporciona mètodes que permeten als objectes executar-se en temps real aprofitant les eines proporcionades pel sistema operatiu. D'altra banda, com s'ha comentat al punt 3.1.2, existeix una interfície anomenada `CNavigationSensor` de la qual tot objecte sensor que proporcionï dades de navegació ha d'heredar. Pel que fa a la generació de fitxers de log, tot objecte capaç de generar dades ha d'heredar d'una interfície anomenada `CLog`, la qual proporciona mètodes per a desar les dades sensorials en fitxers. Finalment, el sensor iGC està dotat amb un detector d'aigua; per a sensors amb aquesta característica, existeix la interfície `CWaterDetector` que proporciona mètodes per a gestionar aquesta característica.

## Implementació

A partir del disseny especificat al punt anterior, obtenim la següent interfície IDL per al nou objecte:

```
#include "../CLog/CLog.idl"
#include "../CNavigationSensor/CNavigationSensor.idl"
#include "../CWaterDetector/CWaterDetector.idl"

struct SFOG{
 string sentence;
 double ifgHeading;
 double igcHeading;
 double pitch;
 double roll;
 double heave;
 long ifgState;
 long driftLock;
 char dataQuality;
 char waterIngress;
};

/// CFOG initialization struct.
struct SFOGInit
{
 /// Here we define the device operation mode, so we know
 /// in what way data we retrieve from the device is formatted.
 /// Notice this parameter only changes software behaviour, not device
 /// operating mode. Device mode can only be modified by hardware.

 string servantName;

 string servantNameNavigator;
 string servantNameDVL;

 SNavigationSensorInit navigationSensor;
```

```

 STRThreadAttributes threadAttr;

 SLogInit logParam;

 string serialPortPath; /// in the robot: "/dev/ttyS1"
};

interface CFOG : CRTPeriodicThread, CLog, CNavigationSensor,
 CWaterDetector
{
 readonly attribute SFOGInit Init;
 readonly attribute SFOG Reading;

 void SyncFogHeadingWithEggCompass();
 void SyncFogHeadingWithDVLCompass();

 double GetFogHeading();
 double GetMagneticHeading();
 double GetPitch();
 double GetRoll();
 double GetHeave();

 double GetHeadingVelocity();
 double GetPitchVelocity();
 double GetRollVelocity();

 double GetFogHeadingOffset();
};

```

Noti's la similitud entre la interfície IDL i la classe C++ dissenyada inicialment. A continuació, a partir d'aquesta interfície IDL, seguirem el mateix procediment explicat al punt REFERENCIA per a obtenir la implementació de l'objecte CFOG.

### 6.2.3 Desenvolupament d'una interfície gràfica per a visualitzar les dades en temps real

#### Motivació

Fins al moment d'iniciar aquest projecte, l'única manera de visualitzar les dades de l'estat de l'Ictineu<sup>AUV</sup> era mitjançant els terminals de consola des dels quals s'executen els diferents scripts encarregats de posar en funcionament tots els objectes de l'arquitectura (vegi's la figura 6.3). Un cop engegats tots els components, aquests imprimeixen línies de text al terminal que informen, entre altres coses, de les mesures preses pels sensors.

Aquestes línies de text poden ser emmagatzemades en un fitxer de text (anomenat log) per ser analitzades off-line amb eines com ara Matlab i realitzar-ne un anàlisi, però no existeix cap forma útil de poder visualitzar les dades en temps real. Per aquesta

```

Terminal — ssh — 80x24
[DVL] Altitude: 11.4184
[DVL] Bottom Velocity (1): [3](0.00690192,0.0179545,0.005)
[DVL] Water Velocity (1): [3](-0.0240981,-0.0357391,-0.009)
[DVL] Iteration
time: 38773.8
[DVL] 2008 9 3 9 17 41 -7.50 -0.10 1.80 1 10.60 0.00 -2.40 1 14.10 14.07 9.70 76
.00 136.00 103.00 100 51.00 0.40 2.80 23.21 1488.00 12.40 2.00 9.70 0.77 -0.26
[DVL] Angle: -0.157077
[DVL] Z: 0.361068
[DVL] Altitude: 11.4407
[DVL] Bottom Velocity (1): [3](0.053,0.0917987,0.024)
[DVL] Water Velocity (1): [3](-0.038366,-0.0644519,-0.018)
[DVL] Iteration
Water Mot: 0 PCs: 0 Bat: 0 Crt: 0
time: 38774.5
[DVL] 2008 9 3 9 17 42 -3.70 -6.80 3.00 1 3.00 0.00 -0.60 1 14.08 14.08 9.70 81.
00 138.00 106.00 100 51.10 0.40 2.80 23.18 1488.00 12.40 2.00 9.70 0.79 -0.26
[DVL] Angle: -0.155332
[DVL] Z: 0.361068
[DVL] Altitude: 11.4376
[DVL] Bottom Velocity (1): [3](0.015,0.0259808,0.006)
[DVL] Water Velocity (1): [3](-0.0773897,0.00195706,-0.03)
[DVL] Iteration

```

Figura 6.3: El robot Ictineu<sup>AUV</sup> operant a la piscina del CIRS.

raó, és interessant disposar d'una interfície gràfica que mostri d'una manera visual i intuïtiva l'estat del robot. Disposant d'una interfície d'aquestes característiques, les tasques realitzades en projectes de recerca o el control del robot durant una missió es faciliten notablement. Així, doncs, s'ha decidit construir una interfície gràfica per als nous sensors de posicionament. De fet, aquesta interfície consistirà en el primer component de l'HMI (Human-Machine Interface) de l'Ictineu<sup>AUV</sup>, un conjunt d'interfícies gràfiques concebudes com a aplicacions independents que permetran visualitzar intuïtivament l'estat del robot i interactuar amb aquest d'una manera accessible a qualsevol persona que hagi mantingut un contacte mínim amb un ordinador.

## Disseny

El disseny de l'interfície gràfica s'ha realitzat tenint la simplicitat com a objectiu primordial. Constarà de tres finestres:

- La finestra principal, mostrada a la figura 6.4. En aquesta finestra es mostrarà, com si es tractés del tauler d'instruments d'una aeronau, un conjunt de tres indicadors gràfics: un per al heading, un per al pitch i un per al roll. Els indicadors de heading i roll consistiràn, cada un, en un cercle amb una escala inscrita que rotarà d'acord amb la mesura rebuda. L'indicador de pitch, com si fos l'indicador de l'alçada de l'horitzó, es desplaçarà amunt i avall d'acord amb el valor rebut. També a través d'aquesta finestra es permetrà configurar el període de temps entre lectura i lectura del sensor.



- La finestra de configuració de connexió. Aquesta finestra permetrà especificar els paràmetres necessaris per realitzar la connexió amb l'objecte corresponent al sensor a través del Naming Service.
- La finestra d'estat de connexió. El procés de connexió amb l'objecte de l'arquitectura O2CA2 corresponent al sensor pot tardar, depenent de l'estat de la xarxa, varis segons. Per a això, aquesta finestra apareixerà a l'usuari amb una barra de progrés i un missatge informatiu que indica l'estat del procés de connexió.

Com s'ha comentat, la interfície permet configurar els paràmetres de connexió amb el Naming Service i també el període de lectura de dades del sensor. És interessant per a l'usuari de l'aplicació que aquests paràmetres es recordin a la màquina on s'executa a fi de no haver d'introduir-los manualment cada vegada, ja que usualment no caldrà canviar-los. Per això, s'ha volgut donar a l'aplicació dissenyada la capacitat d'emmagatzemar dades de configuració en un fitxer XML, que és llegit i modificat convenientment.

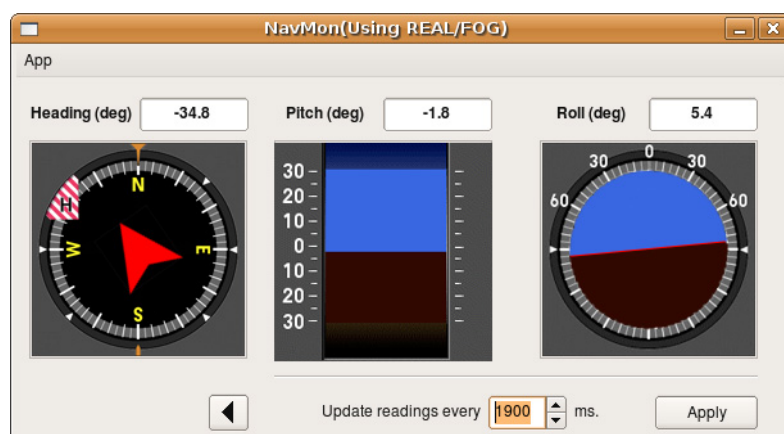


Figura 6.4: Aspecte de la interfície gràfica per als sensors de navegació.

## Implementació

La interfície gràfica s'ha implementat fent servir les següents eines:

- C++ com a llenguatge de programació.
- The ACE ORB (TAO), la implementació de CORBA utilitzada en l'O2CA2 que permet comunicar-nos amb l'objecte CORBA integrat a l'arquitectura que processa les dades dels sensors.
- Qt, les biblioteques d'interfície gràfica més utilitzades per desenvolupar aplicacions en C++.

- Qt Designer, un programari distribuït amb les Qt que permet dissenyar interfícies gràfiques finestra a finestra de manera visual.
- TinyXML, una biblioteca que permet treballar amb documents XML utilitzant C++.

### 6.3 Experiments amb els sensors integrats a l'Ictineu<sup>AUV</sup>

Un cop integrats els nous sensors giroscòpics a l'arquitectura del robot i llesta la interfície per a poder visualitzar les dades d'orientació, els sensors ja poden utilitzar-se en una missió. Durant els dies 11 i 12 de juny de 2008 es van realitzar a Colera (Alt Empordà) uns experiments de reconeixement del fons submarí. Les missions consistien en teleoperar el robot manualment fent-li recórrer trajectòries a una altitud constant mentre es capturaven imatges amb la càmera orientada cap al fons. Un cop finalitzada la missió, a partir de les imatges capturades i de les dades de navegació s'executa un algorisme de visió per computador per a construir un mosaic del fons.

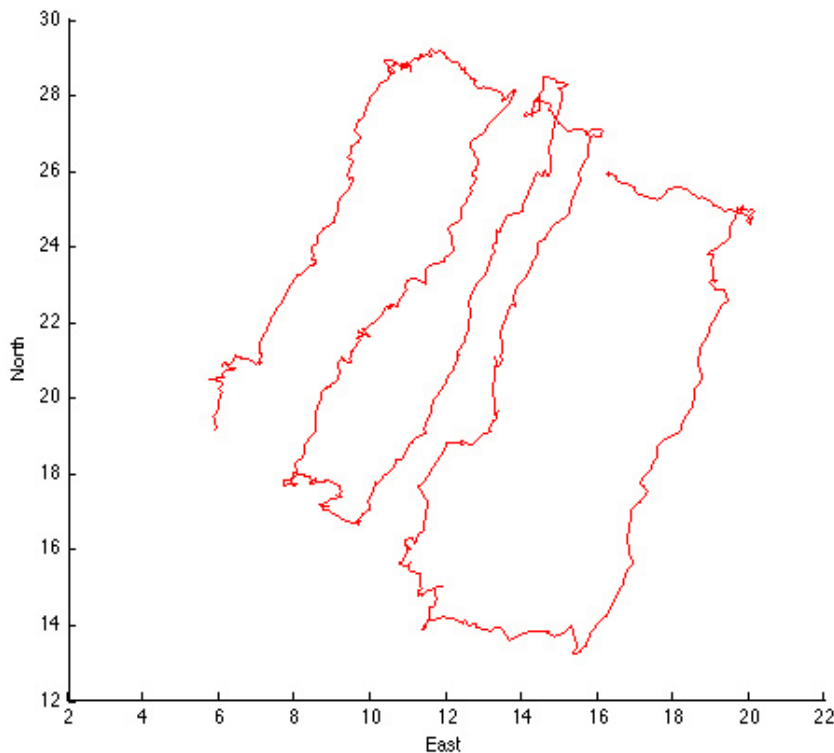


Figura 6.5: Dades de navegació d'una de les trajectòries de reconeixement del fons realitzades a Colera el 12 de juny.

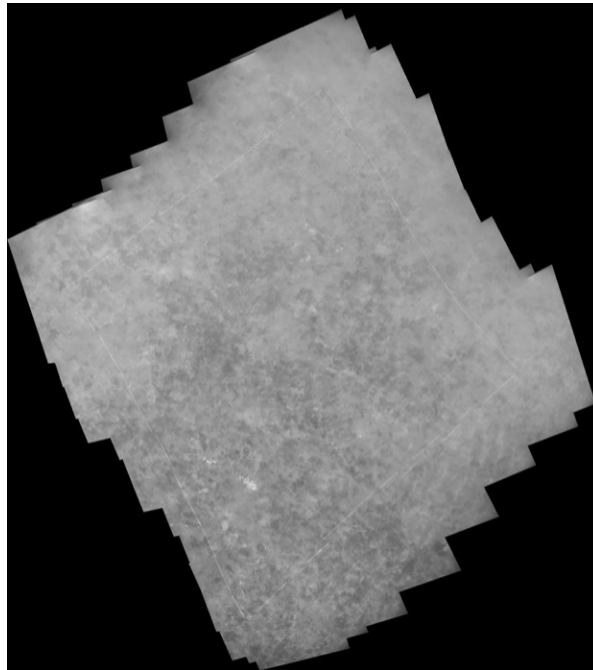


Figura 6.6: Mosaic consttuit a partir de les dades preses amb l'Ictineu<sup>AUV</sup> i els nous sensors giroscòpics a Colera el 12 de juny.



Figura 6.7: El robot operant prop del port de Colera (Alt Empordà).

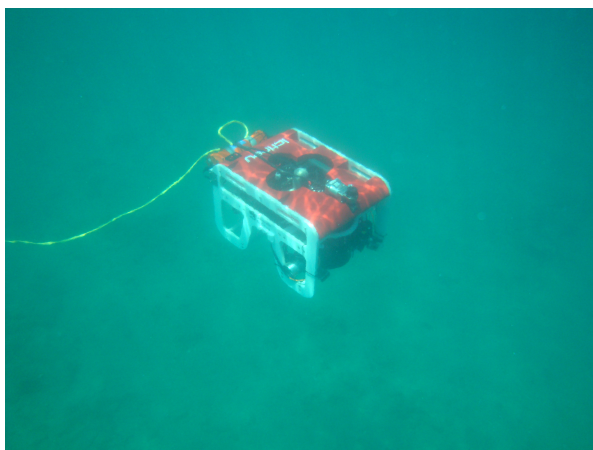


Figura 6.8: Reconeixement del fons marí.

# Capítol 7

## Localització: definició del problema

La segona part d'aquest projecte adreça el problema de la localització d'un robot submarí en un entorn estructurat conegut. Tot i que les idees aquí desenvolupades poden aplicar-se a altres entorns, el treball dut a terme s'ha centrat en realitzar la localització en l'entorn de la piscina del CIRS. Solucionant el problema de la localització per a un robot autònom dins la piscina obre la porta a dur a terme futurs experiments de recerca en condicions de laboratori.

Com s'explicarà, aquest problema ha estat adreçat en el passat per diversos grups d'investigació aplicant diferents mètodes. Aquests mètodes són fruit de l'aplicació de complexos algorismes i eines matemàtiques presents en les investigacions més recents. El treball que es presenta aquí no vol ser part d'aquest model tan teòric i generalista, sinó que busca donar la solució més eficient possible al problema de la localització per al cas particular d'una piscina com la del CIRS aprofitant al màxim la precisió de les mesures proporcionades pels sensors de l'Ictineu<sup>AUV</sup>.

### 7.1 Antecedents

El problema de la localització d'un robot submarí en un entorn estructurat conegut ha estat adreçat per diversos grups de recerca. A l'Autonomous System Laboratory de la Universitat de Hawaii s'utilitzà un mètode consistent en prendre les mesures de distància proporcionades per dos rajos de sonar per a actualitzar un filtre de Kalman [5]. Utilitzant aquest mètode i gràcies a la seva omnidireccionalitat, el robot ODIN<sup>AUV</sup> d'aquesta universitat pot navegar a la vegada que manté la seva orientació relativa respecte les parets de la piscina.

Un sistema més sofisticat es descriu a [1], on un perfilador acústic s'utilitza per traçar les parets de la piscina, permetent al vehicle canviar el seu heading lliurement.

A [4] i [3] es presenten dos mètodes basats en visió per computador. Utilitzant diverses

marques ubicades al fons de la piscina, aquests dos sistemes foren capaços de localitzar el vehicle i de constuir un mapa de marques simultàniament.

En treballs previs, el nostre equip de recerca va solucionar aquest problema utilitzant un patró codificat disposat al fons de la piscina juntament amb un sistema de visió en temps real capaç de proporcionar estimacions de la posició precises a 12 Hz [2].

Més endavant, per prendre part en la competició SAUC-E, es van desenvolupar dos nous algorismes. El primer utilitza un filtre de Kalman extès (EKF) juntament amb un model cinemàtic de velocitat constant, lectures d'un sensor DVL i una brúixola per calcular la trajectòria del vehicle. Per tal de reduir la deriva inherent en aquest procediment, s'utilitzen un sonar i un mapa conegut a priori (definit com un conjunt de línies). En essència, per a cada raig del sonar, es determina un punt d'alta intensitat que hauria de correspondre a un objecte. Després d'un procés d'associació de punt a línia, la discrepància present entre ells s'utilitza per actualitzar la posició del vehicle [?]. Aquest mètode ha de ser inicialitzat amb la posició inicial del vehicle en el mapa, que ha de ser calculada prèviament mitjançant una imatge acústica completa de 360° proporcionada pel sonar. Un altre inconvenient sorgeix quan l'associació de dades falla i la correcció és calculada respecte a un element (línia) equivocat del mapa. Si això passa, el vehicle es perd i l'única manera de relocalitzar-lo és reiniciant el mètode.

El segon algorisme desenvolupat va ser el que finalment es va implementar per a la competició. Bàsicament, consistia en una versió adaptada de la transformada de Hough que utilitzava una imatge acústica completa de 360° i una brúixola per votar per la posició del vehicle en un espai discretitzat que representa l'entorn [?]. La seva simplicitat i el fet que proporciona mesures absolutes de posició eren els seus principals avantatges. D'altra banda, per tal de satisfer els requeriments de temps real, el cost computacional del procés de votació forçà la reducció de la freqüència de producció de mesures així com de la resolució de l'espai.

Finalment, a [?] es presenta un algorisme que pren els avantatges dels dos mètodes acabats de descriure. En aquest algorisme, s'utilitza un EKF per estimar contínuament la posició del vehicle reunint la informació de diversos sensors. Simultàniament, es duu a terme un procés de votació per proporcionar periòdicament correccions lliures de deriva amb mesures absolutes de la posició. Aquest mètode proporciona diversos avantatges. Primerament, l'EKF estima l'estat del vehicle durant el temps entre votacions, reduint la freqüència a la que aquestes correccions de posició han de ser efectuades. D'altra banda, el moviment estimat del vehicle s'utilitza per reduir el moviment induït per les distorsions de les imatges acústiques, obtenint així un procés de votació millorat. A més, el mètode proporciona mesures de posició absolutes en lloc de relatives, permetent conèixer la deriva acumulada i evitant que el vehicle es perdi.

## 7.2 Plantejament de la solució

Tot i que els mètodes de localització que s'acaba de descriure funcionen bé en un marc teòric, tenen inconvenients inherents que els fan poc útils quan operen en un entorn amb les característiques de la piscina del laboratori, lloc on es realitzen la majoria d'experiments. Primerament, el mètode descrit a [5] requereix que es mantingui el heading del robot constant i per tant limita la maniobrabilitat del vehicle. El mètode descrit a [1] es basa en un perfilador acústic, sensor del qual a l'hora d'escriure aquest document l'Ictineu<sup>AUV</sup> encara no disposa. D'altra banda, els mètodes basats en visió per computador impliquen col·locar diverses marques per la piscina, la presència de les quals pot ser molesta per a realitzar alguns experiments. Pel que fa als mètodes més recentment desenvolupats al nostre laboratori, dels quals només l'utilitzat al SAUC-E ha estat implementat, presenten l'inconvenient que es basen fortament en el DVL per a obtenir mesures de posició; el DVL és un sensor basat en ones acústiques i proporciona mesures altament sorolloses quan opera dins la piscina degut a les parets llises de l'entorn. Tot i utilitzar un filtre de Kalman per a fer correccions, les mesures obtingudes són poc precises. D'altra banda, el cost computacional dels mètodes basats en el procés de votació és elevat, i per tant cal reduir-ne la resolució per poder realitzar tasques de control en temps real.

El mètode de localització que s'ha desenvolupat en aquest projecte vol adaptar-se al màxim possible a les característiques de la piscina, mantenint la màxima simplicitat possible per a obtenir un sistema que proporcionï mesures de posició precises en temps real a un cost computacional baix. El disseny es farà aprofitant l'alta fiabilitat de la mesura de heading proporcionada pel nou giròscop de fibra òptica per a associar les dades d'un sonar amb un mapa prèviament conegut. Per a fer això, primer caldrà fer un estudi del funcionament del sonar MiniKing integrat a l'Ictineu<sup>AUV</sup> per després poder dissenyar l'algorisme de localització d'acord amb els coneixements obtinguts.

## Capítol 8

# Estudi del sonar MiniKing de Tritech

El MiniKing de Tritech és un sonar de formació d'imatges compacte dissenyat per utilitzar-se en aplicacions submarines com per exemple detecció d'obstacles o reconeixement d'objectes. Aquest sònar pot dur a terme reconeixements de l'entorn en un pla 2D rotant un raig acústic en forma de ventall a través de petits increments de l'angle del raig. El sonar pot programar-se per cobrir sectors circulars de longitud variable, des d'uns pocs graus fins a escanejos complets de  $360^\circ$ . El raig en forma de ventall, amb un angle d'obertura vertical de  $40^\circ$  i una obertura horitzontal de  $3^\circ$ , permet la formació d'una imatge acústica amb informació suficient sobre l'entorn com per reconèixer mides, formes i reflexions de superfície d'un objecte a distàncies de fins a 50 metres.

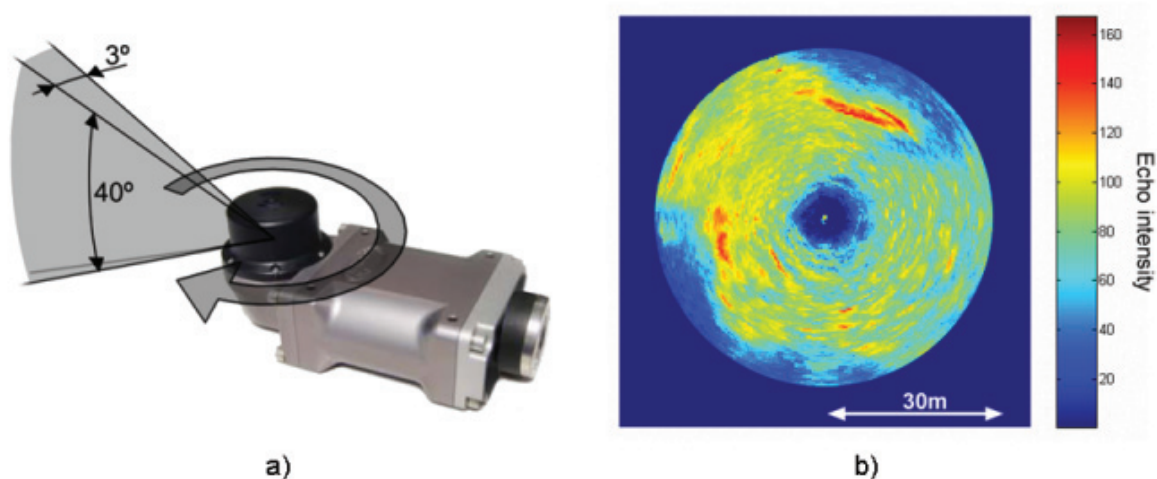


Figura 8.1: a) El sonar MiniKing. b) Imatge acústica adquirida pel sensor.



## 8.1 Interpretació de les imatges capturades pel sonar

Entendre el procés mitjançant el qual es crea la imatge acústica és imprescindible per interpretar la informació obtinguda del sensor. La figura 8.2 mostra el perfil de l'energia de l'eco obtingut a partir d'un raig (*beam*) en forma de ventall que intersecta amb el fons i amb un objecte present a l'entorn.

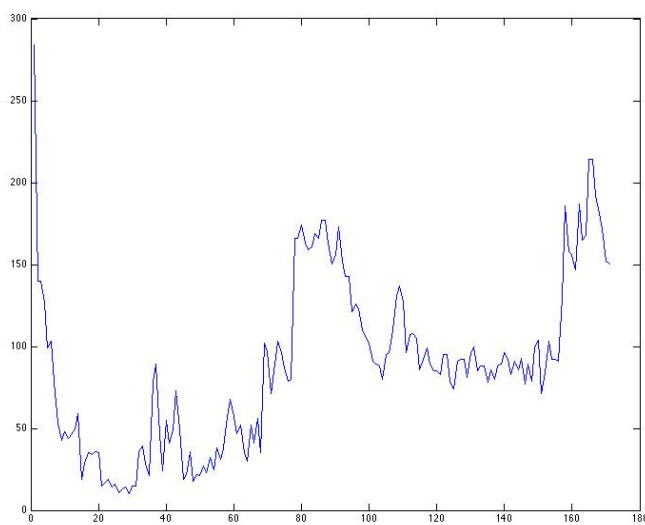


Figura 8.2: Perfil d'energia d'un raig acústic obtingut amb el sonar MiniKing.

El procés comença quan s'emet un senyal acústic o *ping* des del capçal transductor i viatja a través del medi. Quan l'ona col·lisiona amb algun objecte, part de l'energia és retornada. Per tant, si es coneix el temps transcorregut durant el vol de l'ona, es pot inferir la distància a l'objecte. A més, analitzant la intensitat del so retornada, poden determinar-se algunes característiques de l'objecte. Tota aquesta informació és donada pel sensor com un vector de valors d'intensitat, anomenats *bins*, cada un corresponent a la distància determinada per la resolució amb la qual s'ha configurat el sonar.

A partir de la conjunció de la informació proporcionada per diversos *beams*, pot formar-se una imatge completa de l'entorn del sensor. Si el capçal transductor rota amb petits increments regulars d'angle i s'emeten *pings* successius, llavors la informació rebuda pot processar-se per obtenir una representació completa del que hi ha present al voltant del sistema en un radi determinat pel rang al qual s'ha configurat el sonar.

Habitualment, la imatge acústica obtinguda s'assembla d'alguna manera a una imatge òptica obtinguda del mateix escenari. Tot i això, una imatge acústica real és sovint més difícil d'interpretar per diverses raons. Primer, una imatge de sonar sempre tindrà una

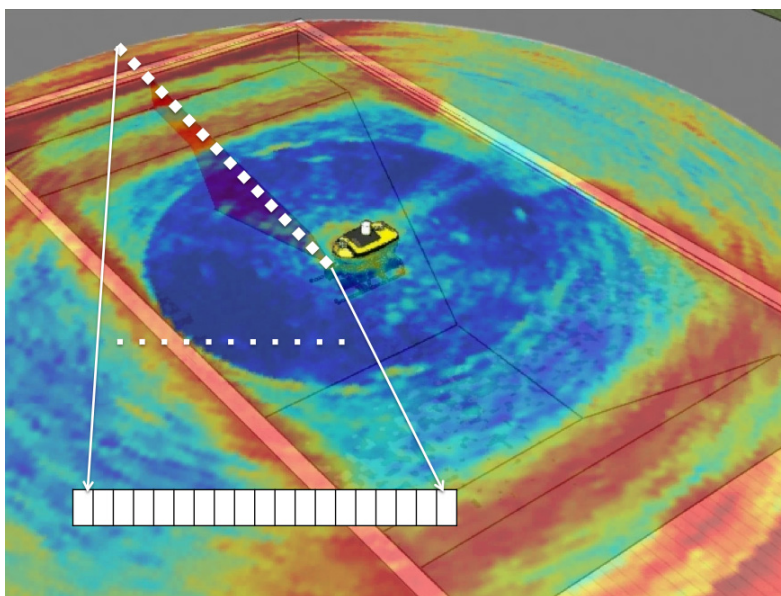


Figura 8.3: Adquisició d'un raig acústic.

resolució baixa deguda a la naturalesa dels senyals acústics utilitzats per generar-la. A més, els materials que conformen l'entorn són determinants a l'hora d'obtenir informació del sonar. Generalment, els objectes rugosos són millors objectius per a un sonar perquè retornen ecos en diverses direccions. D'altra banda, les superfícies llises poden provocar una forta reflexió en una direcció particular, i pràcticament cap en qualsevol altra.

## 8.2 Paràmetres de configuració

El sonar MiniKing permet a l'usuari configurar un ampli ventall de paràmetres, dels quals resulten interessants els següents:

- **Rang.** És el paràmetre que regula la distància màxima a la qual arriben les ones acústiques d'un raig. Es dona en decímetres.
- **Resolució.** Determina el la longitud d'una mostra d'intensitat, es dona en decímetres.
- **Step size.** Determina l'increment d'angle entre raig i raig. Es dona en graus.

### 8.3 Desenvolupament d'una interfície gràfica per a la visualització de les imatges acústiques

Tot i que el sonar MiniKing és un sensor que ja està integrat a l'arquitectura de control O2CA2, l'única manera d'analitzar les dades obtingudes amb el dispositiu és desant les mesures en fitxers durant una missió i analitzant-los posteriorment. És interessant, doncs, poder disposar d'una interfície gràfica que mostri les imatges acústiques obtingudes pel MiniKing en temps real, mentre el robot és teleoperat. Disposant d'aquesta eina, serà molt més fàcil entendre la informació rebuda i dissenyar un sistema de posicionament que la utilitzi.

L'aplicació gràfica s'ha dissenyat similarment a l'aplicació construïda anteriorment per visualitzar els sensors de posicionament relatiu. El disseny consistirà també en tres finestres, de les quals només variarà significativament respecte l'altra interfície la finestra principal, mostrada a la figura 9.6.

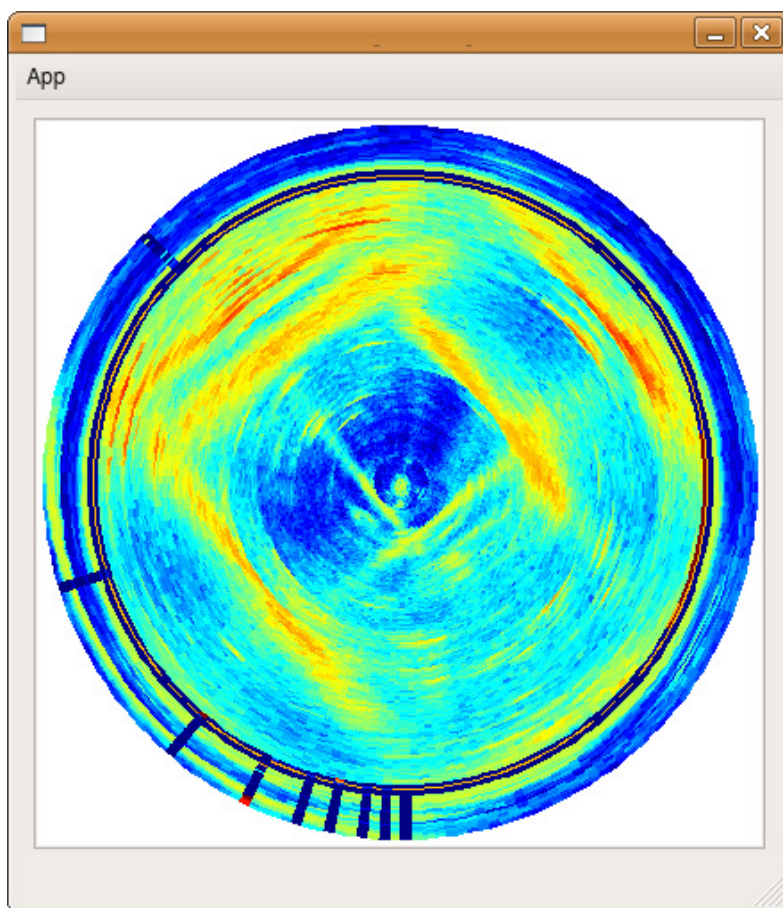


Figura 8.4: Interfície gràfica per a la visualització d'imatges acústiques en temps real.

## Capítol 9

# Disseny del sistema de localització

Segons les especificacions plantejades al capítol REFERENCIA, en aquest capítol s'explica el procés que s'ha seguit per construir un sistema de localització basat en un giròscop de fibra òptica i un sonar. Aquest sistema s'ha dissenyat per funcionar en un entorn estructurat, i s'ha desenvolupat tenint la piscina del CIRS com a referència. Aquest sistema de localització s'ha ideat com el punt de partida per a poder disposar d'un sistema de navegació robust que permeti realitzar experiments de manera àgil al laboratori.

### 9.1 Entorn de treball

El sistema de localització s'ha dissenyat per a funcionar bé dins la piscina del laboratori. Per tant, primer que cal fer és conèixer l'entorn on haurà de funcionar el sistema desenvolupat. A continuació es descriu l'edifici del CIRS i les eines de les quals s'ha disposat allà per a desenvolupar el sistema.

El centre de recerca consta d'una piscina i d'un centre de supervisió i control situat sota el nivell de l'aigua, on pot observar-se l'entorn subaquàtic a través d'un vidre. Sobre el nivell de l'aigua, hi ha una sala amb sortida directa a la piscina que fa de garatge per al robot IctineuAUV. Des d'aquesta sala, amb l'ajuda d'un petit elevador mecànic, s'entra i es treu el robot de l'aigua. La piscina, que pot veure's a la figura 9.1 mesura 16m de llarg, 8 d'ample i té dues rampes simètriques que comencen a un metre de profunditat i descendeixen fins a un màxim de 5m.

La piscina és un entorn complicat per operar-hi amb dispositius acústics. Es tracta d'un espai tancat, i les seves parets tenen una alta reflectivitat. Per aquestes raons, quan es treballa amb el sonar és fàcil trobar-se amb reflexes fantasma a les imatges acústiques obtingudes. Aquests reflexes apareixen quan passa esgüent: el senyal acústic avança fins que impacta amb la paret. Això provoca un canvi en el perfil d'intensitat de l'ona acústica. Després d'aquesta paret, cap altre objecte hauria d'aparèixer a la imatge. Però

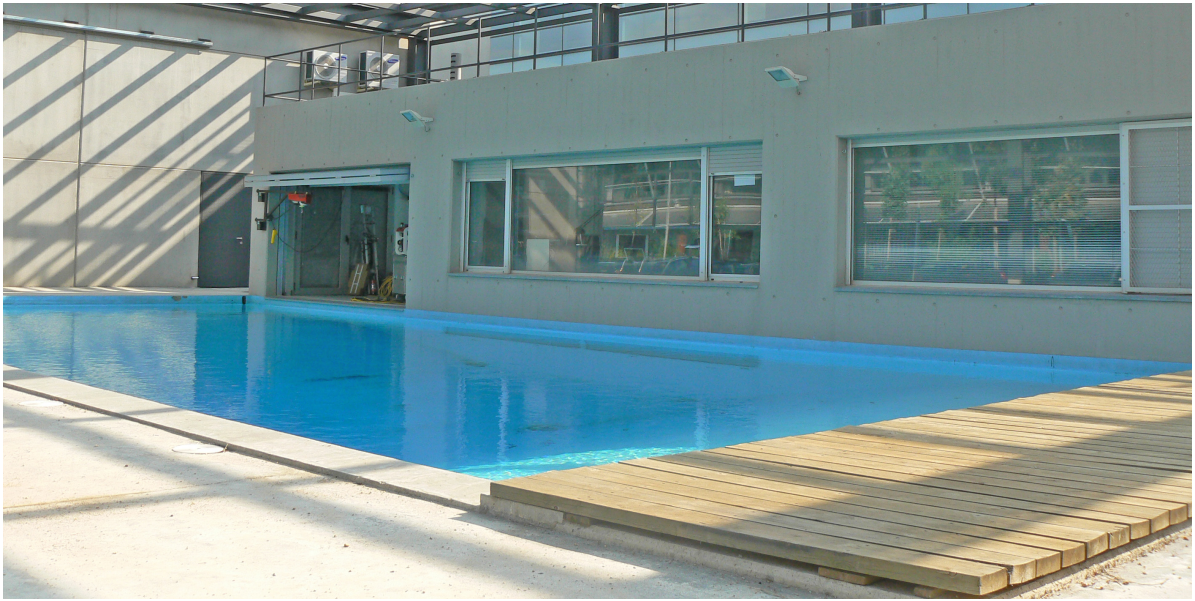


Figura 9.1: Edifici i piscina del CIRS.

degut a la suavitat de la superfície, si el sonar s'ha configurat per emetre els pings a prou distància com perquè el raig reboti a la primera paret i arribi a la que hi ha davant, obtindrem un segon canvi en el perfil d'intensitat. Per tant, s'estarà detectant un objecte que no existeix realment. Caldrà tenir en compte tot això a l'hora de dissenyar el sistema de localització.

## 9.2 Disseny de l'algorisme

El mètode de posicionament que s'ha dissenyat està pensat per a funcionar sobre el pla 2D de la piscina que està més aprop de la superfície, aproximadament fins a dos metres de profunditat. En aquesta zona la forma de la piscina és regular, podent-se modelar com un rectangle 2D de  $16 \times 8$  m. Per tant, inicialment l'algorisme s'ha treballarà en aquest pla utilitzant geometria euclidiana per associar les imatges acústiques captades pel sonar al model de la piscina d'acord a la mesura de heading proporcionada pel giròscop de fibra òptica.

### 9.2.1 Idea general

La idea de l'algorisme és la següent. El sonar es configura de manera que va escombrant cercles complets de  $360^\circ$  contínuament. Coneixent la orientació del mapa (la piscina) i el vehicle respecte el nord, pot calcular-se la orientació absoluta respecte el nord de cada raig proporcionat pel sonar. Es sap també que el pla horitzontal de la piscina



és una estructura poligonal, concretament rectangular. Per tant, quan un raig acústic incideix perpendicularment en un dels costats del rectangle (una paret), es pot conèixer la distància a aquest costat. De fet, no cal que el raig hi incideixi perpendicularment; l'únic que cal és evitar prendre rajos que incideixen a les cantonades del rectangle, ja que no és possible llavors distingir a quin costat han incidit. Així, doncs, coneixent la orientació del raig acústic i de la distància a la paret que proporciona, es pot saber a quina distància i de quina paret es troba el vehicle per a cada raig. Això es farà servir per a actualitzar la posició del robot en cada una de les dues dimensions del mapa.

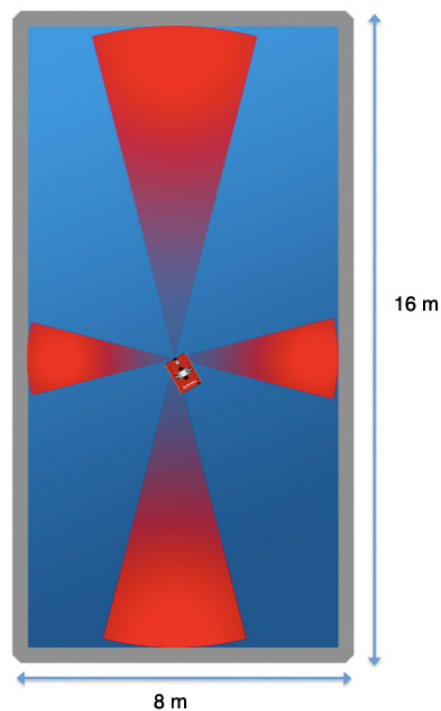


Figura 9.2: Esquema de funcionament de l'algorisme de localització.

### 9.2.2 Sistemes de coordenades

En el sistema de navegació que s'està desenvolupant hi intervenen tres sistemes de coordenades. El primer, anomenat sistema de coordenades de sensor, està situat sobre el sonar tal com es mostra a la 9.3. El sistema de coordenades de robot, també mostrat a la figura 9.3, es situa al centre de gravetat del vehicle. Finalment, el sistema de coordenades de món es situarà a la cantonada de la piscina on hi ha l'elevador mecànic que permet ficar el robot a l'aigua, ja que és allà on comencen la majoria de missions (veure figura 9.4).

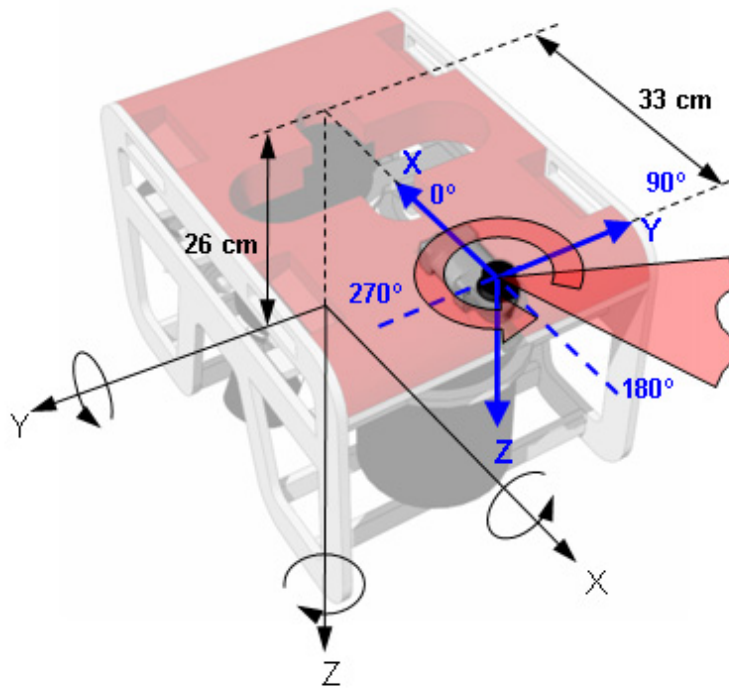


Figura 9.3: Sistema de coordenades de robot i sistema de coordenades del sonar. Pot observar-se el sentit de rotació mecànica del sonar quan realitza els escombrats, i l'escala angular que fa servir.

### 9.2.3 Configuració del sonar

La configuració del sonar és un punt crític per al correcte funcionament de l'algorisme. El primer paràmetre que cal ajustar és el rang: caldrà optimitzar-lo de manera que els rajos acústics tinguin prou energia per recórrer la distància que separa els dos punts més allunyats de la piscina. El valor però haurà de ser el més petit possible per minimitzar l'aparició de reflexes fantasma. Com es mostra a la figura X, pot calcular-se la distància màxima  $d_{max}$  entre els punts més allunyats de la piscina calculant la distància euclidiana entre els punts  $(0, 0)$  i  $(16, 8)$  del pla cartesià que representa el mapa del món:

$$d_{max} = \sqrt{(16 \times 16) + (8 \times 8)} = 17.88 \simeq 18 \text{ m.}$$

Per tant es configurarà un rang de 18 m. Els dos altres paràmetres que cal configurar són la resolució i l'stepsize, que determinaran la precisió màxima del sistema de localització. Per tant, seleccionarem els mínims valors possibles: 1 dm. de resolució i un stepsize de  $1.8^\circ$ . Amb aquesta configuració s'obtindrà el major nombre de rajos per volta amb el major nombre de mesures d'intensitat.

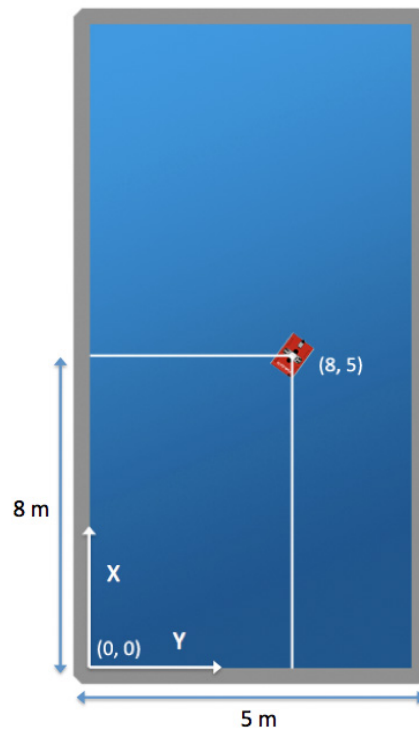


Figura 9.4: Sistema de coordenades de món i exemple de posició del robot.

#### 9.2.4 Segmentació de les imatges acústiques

Sabem que per a cada bin (valor d'intensitat d'un raig), el sonar retorna un valor de 0 a 255. Però els valors baixos no resulten interessants, ja que no corresponen a la presència d'un objecte. Per tant, cal segmentar els rajos de manera que s'obtingui una imatge acústica amb només les correspondències amb un objecte. Per fer això es descartarà tots els bins menors que un cert llindar. Experimentalment s'ha comprovat que un valor al voltant de 160 segmenta bé la imatge deixant només les zones d'interès. A més, també es descartarà tots els rajos que no estiguin dins les zones especificades a la figura 9.2, així com els valors corresponents al primer metre del rang, que com pot veure's a la figura 8.2 corresponen a valors sorollosos que distorsionen la mesura de la distància. A la figura 9.5 es mostra una imatge acústica corresponent a un escombrat sencer i la mateixa imatge havent-hi aplicat el procés de segmentació.

#### 9.2.5 Càlcul de la distància i actualització de la posició

L'actualització de la posició del robot no es realitzarà per a cada beam llegit del sonar, sinó per a cada una de les zones mostrades a la figura 9.2. És a dir, per a cada raig acústic es realitzarà el procés de segmentació. Tot seguit, havent eliminat els valors



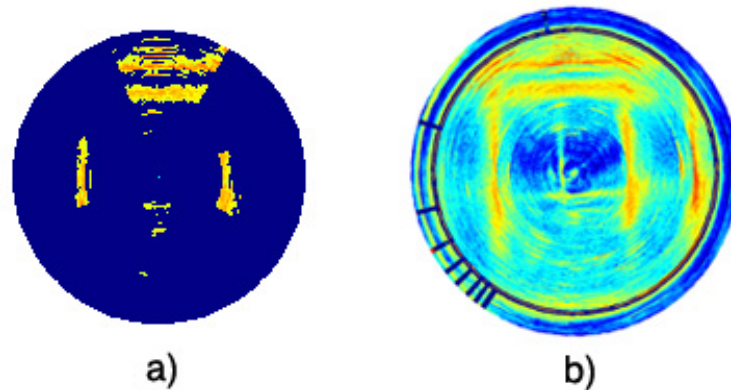


Figura 9.5: Segmentació d'una imatge acústica. a) Imatge original. b) Imatge amb el procés de segmentació aplicat.

que no són d'interès, es cercarà el punt d'intensitat més alt d'aquest raig. Sabent la posició del punt d'intensitat més alt, es coneix la distància a l'objecte. A continuació cal projectar aquesta distància sobre l'eix  $X$  o  $Y$  del sistema de coordenades de món, depenent de l'angle del raig acústic.

Un cop realitzat aquest procés es desarà la distància calculada en un *buffer*. Això es repetirà successivament durant l'escombrat del sonar per a cada raig acústic que es trobi dins una zona vàlida. Finalment, quan arriba un raig que és fora de la zona es calcula la mitjana dels valors desats al *buffer* i s'actualitza la posició del robot.

Resumint, l'actualització de la posició es duu a terme a partir dels següents passos:

- Per a cada raig acústic dins una zona vàlida:
  - Prendre les mesures d'orientació del robot i l'angle del raig.
  - Realitzar els canvis de sistemes de coordenades per tal d'obtenir la orientació del raig relativa al sistema de coordenades de món.
  - Segmentar el vector de valors intensitats corresponent al raig.
  - Calcular el valor màxim del vector d'intensitats i obtenir així la distància a l'objecte (la paret).
  - Desar la distància calculada al *buffer*.
- Si el darrer raig acústic estava en una zona vàlida i el que ha arribat en el present instant no està a cap zona vàlida, es calcula la mitjana dels valors del *buffer* i s'actualitza la posició en  $X$  o en  $Y$ , depenent de la zona on es trobessin els valors del *buffer*.

### 9.3 Implementació

Per a implementar l'algorisme de localització, s'ha desenvolupat una aplicació GUI que permet visualitzar les següents dades en temps real:

- La imatge acústica segmentada amb un llindar, que pot modificar-se *onthe fly* mentre s'executa l'aplicació.
- Els valors de posició en  $X$  i  $Y$ .
- La orientació relativa del robot respecte el sistema de coordenades de món.
- El temps transcorregut des que s'ha començat a executar l'algorisme.
- La configuració del sonar, incloent rang resolució i step size.

L'aplicació s'ha desenvolupat seguint el mateix procediment ja explicat per a les interfícies dels sensors giroscòpics i del sonar mateix. De fet, s'ha realitzat una ampliació de la interfície del sonar afegint-hi l'algorisme implementat utilitzant CORBA i C++ i els widgets necessaris per a mostrar les dades que s'acaba de comentar. L'aplicació GUI per al sistema de localització té l'aspecte mostrat a la figura ??.

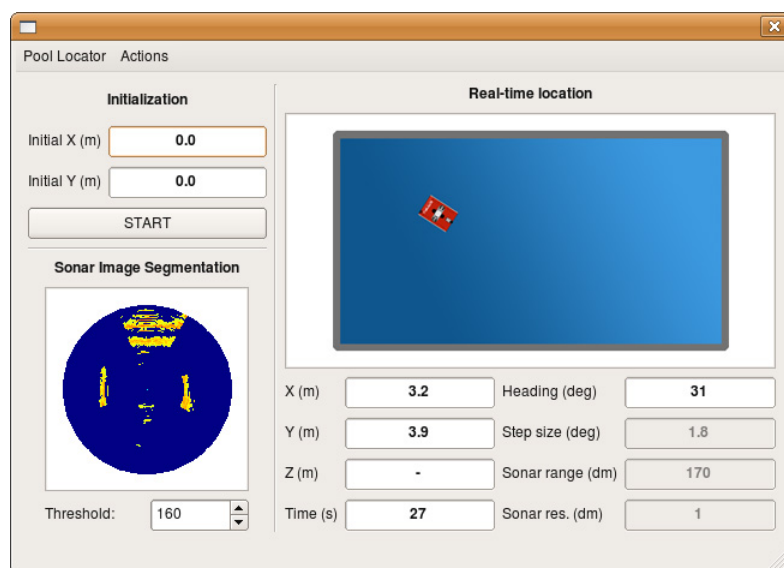


Figura 9.6: Interfície gràfica del sistema de localització executant-se en temps real.

Fent ús d'aquesta aplicació, s'ha procedit a comprovar la precisió de l'algorisme.

## 9.4 Disseny i realització dels experiments de precisió

Un cop desenvolupat el sistema de localització, cal estudiar-ne la seva precisió. D'entrada, cal tenir en compte que el sistema no serà mai més precís que les mesures proporcionades pels sensors. Pel que fa al sonar, això implica que no es podrà obtenir una precisió superior als 10 centímetres de resolució màxima que ofereix el sensor. D'altra banda, la deriva dels sensors giroscòpics també afectarà la precisió dels càlculs, amb més severitat conforme transcórrer el temps.

Tenint clar això, es posarà a prova el sistema de localització realitzant dos experiments:

- El primer consisteix en fixar el robot en una posició coneguda de la piscina, mesurada amb una cinta mètrica. A continuació s'executarà el sistema de localització durant uns segons i es desarà les dades obtingudes en un fitxer de log per al posterior anàlisi. L'experiment es realitzarà en cinc posicions diferents del pla de la piscina; aquestes seràn (4, 2), (4, 6), (12, 6), (12, 2) i (8, 4). Com es pot observar, quatre d'aquestes posicions són properes a les cantonades de la piscina i la darrera correspon al centre de la mateixa.
- El segon consistirà en realitzar una trajectòria rectangular. Aquesta recorrerà els mateixos punts utilitzats en l'experiment anterior (excepte el punt central): (4, 2), (4, 6), (12, 6), (12, 2). Igualment es desarà les dades obtingudes en un fitxer de log que s'analitzarà. A la figura 9.7 pot veure's l'Ictineu<sup>AUV</sup> iniciant la trajectòria.

## 9.5 Resultats

Les figures 9.8, 9.9, 9.10, 9.11 i 9.12 mostren les dades de localització obtingudes mantenint el robot estàtic, respectivament, a les posicions (4, 2), (4, 6), (12, 6), (12, 2) i (8, 4). La figura 9.13 mostra una gràfica amb les dades de localització obtingudes fent la trajectòria.

A partir dels gràfics obtinguts, pot inferir-se que el sistema de precisió desenvolupat té una precisió d'aproximadament entre 1 i 2 metres. Tot i això, la presència de soroll en les imatges acústiques provoca l'aparició de posicions fantasma, on realment el robot no es troba. Tot i això, als gràfics pot veure's una densitat de punts considerablement alta en les zones on el robot es trobava realment durant la realització dels experiments.



Figura 9.7: El robot al primer punt fixat per als experiments de localització.

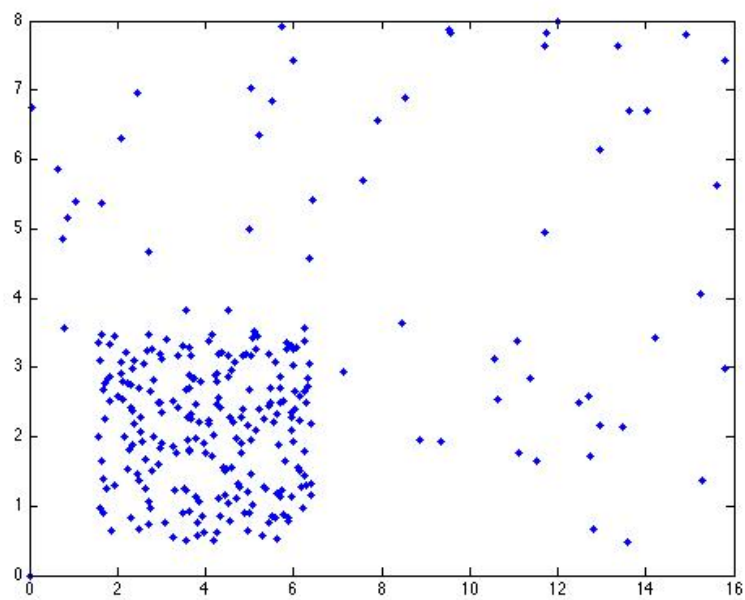


Figura 9.8: Resultat per al punt (4,2).

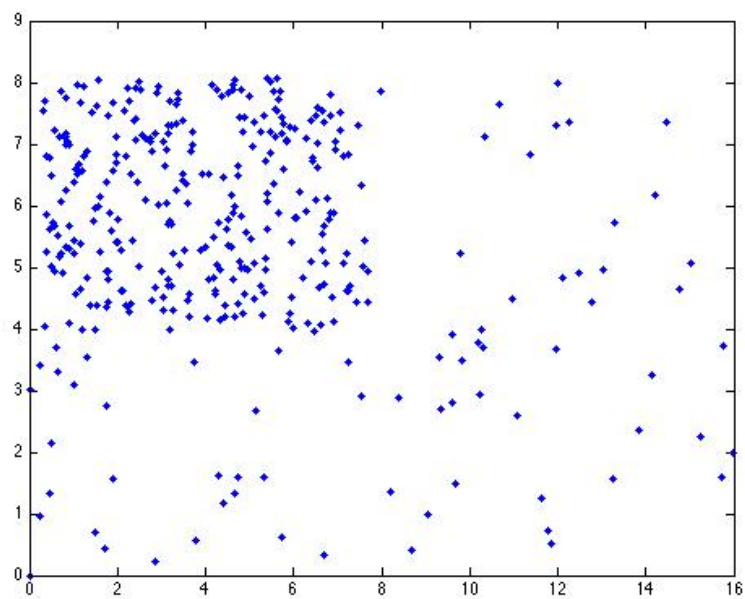


Figura 9.9: Resultat per al punt (4,6).

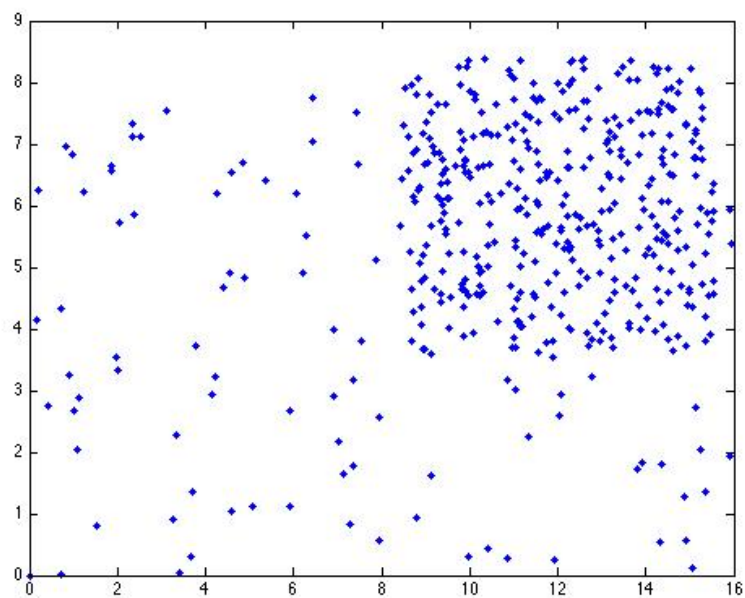


Figura 9.10: Resultat per al punt (12,6).

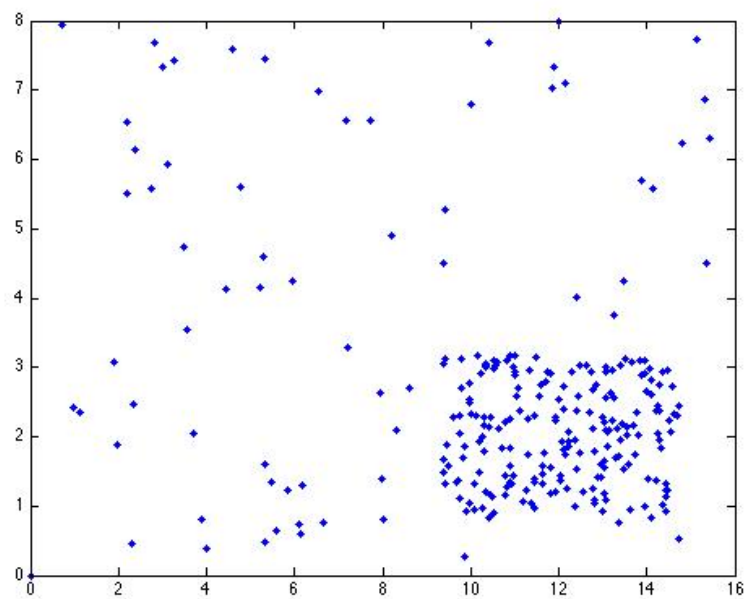


Figura 9.11: Resultat per al punt (12,2).

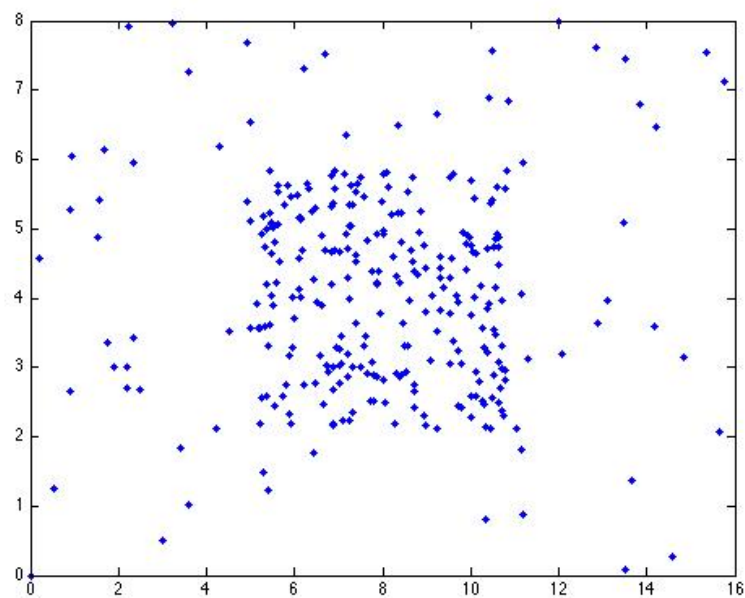


Figura 9.12: Resultat per al punt (8,4).



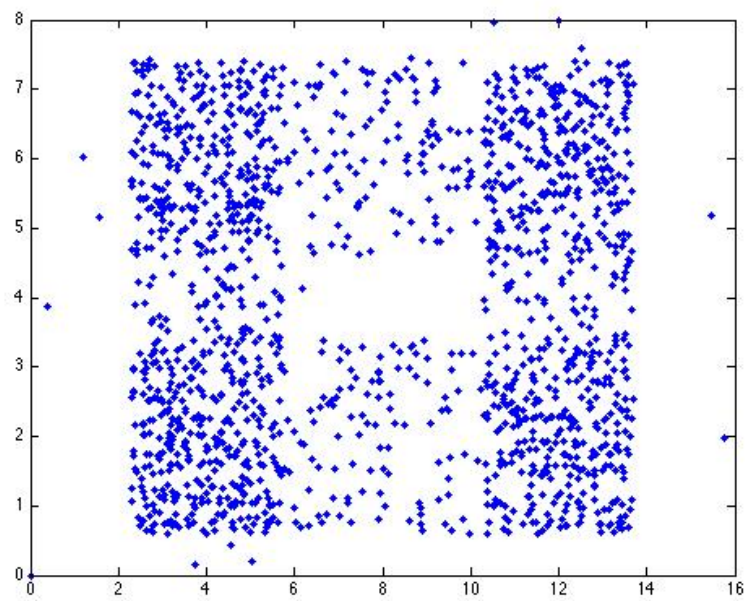


Figura 9.13: Resultat obtingut amb la trajectòria feta pels punts  $(4,2)$ ,  $(4,6)$ ,  $(12,6)$  i  $(12,2)$ .

# Capítol 10

## Conclusions i treballs futurs

Arribats a aquest capítol, donarem una ullada a la multitud de tasques dutes a terme durant la realització d'aquest projecte. Es va començar fent una primera presa de contacte amb el sistema de sensors giroscòpics que ha estat la peça central de tota la feina feta: el conjunt format per l'iGC i l'iGC de Tritech. Va caler llegir la documentació que acompanyava els aparells així com cercar informació sobre els principis físics en els quals es basen. Així, es va fer un estudi detallat dels sensors, realitzant experiments per a comprovar-ne el funcionament sota l'efecte de camps magnètics provocats per cossos metàl·lics com per determinar-ne la deriva.

Un cop coneguts els sensors i comprès el seu funcionament i particularitats, es va procedir a desenvolupar una aplicació en C++ que processés les dades dels sensors giroscòpics. El desenvolupament d'aquesta aplicació va permetre realitzar l'experiment de comprovació de la sortida dels sensors. Realitzant aquest experiment, es van descobrir incoherències entre la documentació dels formats de trama i la sortida real provinent dels sensors. Arran d'aquest fet, es va contactar amb els enginyers de Tritech, que van solventar l'error proporcionant una documentació actualitzada.

Tot seguit es va iniciar l'estudi de l'arquitectura O<sup>2</sup>CA<sup>2</sup> a fi de poder desenvolupar un objecte integrat en aquesta arquitectura que permetés processar les dades d'orientació proporcionades pels sensors i deixar-les disponibles per a tot el programari de control de l'Ictineu<sup>AUV</sup>. L'estudi d'aquesta vasta peça de software ha estat realment un repte, tant per la seva enorme jerarquia de més de trenta classes com per el gran nombre de tecnologies que hi prenen part: CORBA, C++, aplicacions de temps real, transmissions sèrie, computació distribuïda... i un llarg etcètera.

Coneixent ja el funcionament del maquinari dels sensors i de l'arquitectura de control, es començà a desenvolupar el programari necessari per a afegir el nou dispositiu sensorial a l'arquitectura. Aquest procés va implicar el disseny d'un objecte CORBA degudament integrat en l'estructura de l'arquitectura O<sup>2</sup>CA<sup>2</sup>. Al mateix temps, es va determinar juntament amb els enginyers industrials i electrònics del laboratori encarregats del manteniment de l'Ictineu<sup>AUV</sup> com realitzar la instal·lació física dels sensors a



l'estructura del robot.

Ja amb els sensors integrats a l'Ictineu<sup>AUV</sup> i funcionant a ple rendiment, es va dissenyar una interfície gràfica per a la visualització de la informació sensorial del sistema giroscòpic. Per a desenvolupar aquesta aplicació gràfica, va caler enfrontar-se a un nou repte: les biblioteques Qt i TinyXML. Fou necessari, doncs, llegir i comprendre la documentació d'aquestes biblioteques per a poder implementar l'aplicació. Mitjançant aquesta interfície, ara és possible teleoperar el robot tenint una imatge intuitiva de la seva orientació.

Havent integrat els giròscops tant al programari com al maquinari del robot, el nou desafiament consistí en el disseny d'un sistema de localització 2D que permetés aprofitar les capacitats dels nous sensors. Per a fer això, primer va caler estudiar el funcionament del sonar MiniKing, l'altra peça clau per a aquest sistema de localització. Va caler, doncs, fer nova recerca comprènent la documentació del sensor i llegint sobre el funcionament d'aquests dispositius. Un cop comprès tant el MiniKing com l'objecte sensor de l'arquitectura O<sup>2</sup>CA<sup>2</sup> que el gestiona, es va procedir a desenvolupar una interfície gràfica (aprofitant parcialment la feina feta en l'anterior aplicació GUI per als giròscops) que permetés visualitzar els imatges acústiques capturades pel sonar. El desenvolupament d'aquesta interfície va permetre estudiar les imatges acústiques i començar a formar idees sobre com hauria de ser un sistema de localització basat en un giròscop i un sonar.

A partir d'aquí es va iniciar el disseny del sistema de localització 2D. L'objectiu d'aquest sistema és establir les bases per aconseguir disposar d'un sistema de control que permeti operar a l'Ictineu<sup>AUV</sup> en un entorn tant complicat per a un robot submarí com és la piscina del laboratori. Per a construir el sistema es va idear un mètode geomètric basat en transformacions homogènies, canvis de sistemes de coordenades i trigonometria, prescindint de les complicades eines matemàtiques pròpies de la més recent recerca. Aquestes estàn pensades per a casos molt generals i resulta molt difícil aplicar-les al cas de la piscina del laboratori. Un cop dissenyat el mètode, es va implementar creant una nova aplicació gràfica basada en CORBA i Qt que interactua amb l'arquitectura de control i executa l'algorisme desenvolupat per a determinar la posició del robot dins la piscina.

Finalment, s'ha realitzat experiments per a comprovar la precisió del sistema de localització desenvolupat i s'ha documentat la tota la feina feta en aquesta memòria. Per fer-ho, s'ha afrontat el repte d'aprendre el funcionament del sistema L<sup>A</sup>T<sub>E</sub>X d'escriptura de documents científics, amb la finalitat que serveixi d'ajuda per a la redacció de futurs documents i volent obtenir una presentació pulcre i ben acabada.

## 10.1 Conclusions

Responent en ordre els objectius plantejats al punt 1.2, podem concluir el següent:

- **Estudi del funcionament d'un conjunt sensorial basat en una unitat de referència inercial i un giròscop de fibra òptica.** S'ha realitzat aquest estudi satisfactòriament, proporcionant al CIRS la informació necessària per a integrar el sensor al robot Ictineu<sup>AUV</sup> i per a utilitzar el sensor en futures aplicacions.
- **Estudiar l'arquitectura de control del robot Ictineu<sup>AUV</sup> i comprendre el seu funcionament amb la finalitat de construir peces de programari que n'ampliïn la funcionalitat.** S'ha dut a terme l'estudi i s'ha comprès el disseny de l'arquitectura de control O<sup>2</sup>CA<sup>2</sup>, fet que queda demostrat amb l'assoliment del següent objectiu.
- **Desenvolupament del programari necessari per integrar aquests sensors a l'arquitectura de control del robot Ictineu<sup>AUV</sup>.** No sols s'ha desenvolupat el programari que ha permès integrar els sensors giroscòpics al robot, sinó que, a més, s'ha creat una aplicació gràfica que permet la visualització de les dades proporcionades pels sensors en temps real. Queda, doncs, plenament assolit aquest objectiu i es demostra que la feina esmentada als dos primers punts d'aquestes conclusions ha donat el seu fruit.
- **Comprovació del correcte funcionament i la precisió del conjunt sensorial mitjançant experimentació.** S'ha dut a terme experiments diversos per a comprovar el funcionament, sensibilitat i deriva del sistema giroscòpic estudiat. Els resultats obtinguts han permès formar una idea del rendiment real dels aparells i han fomentat un contacte amb el fabricant per tal d'obtenir-ne un funcionament òptim.
- **Estudi del funcionament del sonar integrat al robot Ictineu<sup>AUV</sup>.** S'ha realitzat satisfactòriament l'estudi, construint a partir d'aquest una segona aplicació gràfica que permetrà visualitzar les imatges acústiques de l'entorn del robot mentre aquest realitza missions ja siguin teleoperades o autònomes. A més, a partir de la interfície desenvolupada s'ha pogut completar l'estudi podent comprovar en temps real el que el robot *veu* en diferents posicions de la piscina, prenent així idees per a atacar el següent objectiu.
- **Disseny d'un sistema de localització basat un sonar i un giròscop de fibra òptica que permeti determinar la posició del robot a la piscina en temps real.** S'ha ideat un mètode de localització geomètric, basat en les característiques particulars de la piscina del laboratori. Aquest sistema es basa en la feina feta per a integrar els sensors giroscòpics i en l'estudi del sonar realitzat.
- **Implementació el sistema de localització.** S'ha implementat el mètode dissenyat i s'ha desenvolupat una interfície GUI que permet fer un seguiment de l'execució de l'algorisme, visualitzant les imatges acústiques i les dades d'orientació i localització. S'ha acomplert, doncs, aquest objectiu.

- **Comprovació del correcte funcionament i la precisió del sistema de localització desenvolupat.** S'ha dut a terme experiments per a determinar la precisió del sistema desenvolupat i se n'ha analitzat el rendiment, com s'explica al capítol 9.

## 10.2 Treballs futurs

Abans de començar a citar algunes de les futures aplicacions o ampliacions de la feina realitzada en aquest projecte, cal recordar que el sistema de localització desenvolupat en la segona part d'aquest no vol ser la solució definitiva a la navegació del robot Ictineu<sup>AUV</sup>, sinó establir la base per a seguir treballant en aquesta línia a fi de poder disposar d'un sistema de navegació robust i que treballi bé en el complicat entorn de la piscina del laboratori. Un cop comentat aquest detall, procedim a enumerar possibles treballs futurs basats en aquest projecte:

1. Obtenir del fabricant Tritech la informació necessària per a poder calibrar correctament el sensor iGC. Cal tenir en compte que aquesta tasca no és tan trivial com pot semblar, ja que implica accedir a l'interior del robust encapsulat del sensor (pensat per a submergir-se fins a 4000 m.) i interactuar amb la seva circuiteria interna.
2. Per al sistema de localització desenvolupat, pot ser interessant aplicar-hi algun sistema de tracking en les mesures. Aquest sistema, basat en la presa de decisions d'acord amb un vector de les darreres posicions obtingudes, podria ajudar a reduir l'aparició de posicions fantasma i d'errors en les execucions del sistema, fent-lo menys sorollós i més robust.
3. El sistema de localització desenvolupat funciona en el pla horitzontal del robot i s'ha dissenyat per a executar-se en la zona més propera a la superfície. Partint d'aquest mètode geomètric 2D, pot utilitzar-se el sensor de pressió que incorpora el DVL a fi de determinar la profunditat a la que es troba el robot i per tant obtenir una localització 3D. Caldria però tenir molt en compte la forma irregular de la piscina del laboratori.

# Bibliografia

- [1] M. Caccia, G. Bruzzone, and G. Veruggio. Sonar-based guidance of unmanned underwater vehicles. *Advanced Robotics*, 15(5):551–573, September 2001.
- [2] M. Carreras, P. Ridao, J. Batlle, and D. Ribas. High-accuracy localization of an underwater robot in a structured environment using computer vision. In Springer Berlin/Heidelberg, editor, *Iberian Conference on Pattern Recognition and Image Analysis. Lecture Notes in Computer Science*, volume 2652, pages 150–157, 2003.
- [3] G. García de Marina, D. Ribas, and P. Ridao. A global localization system for structured environments using an imaging sonar. In *Proceedings of the IFAC Conference on Control in Marine Systems*, Bol, Croatia, September 2007.
- [4] J. H. Lim and J. J. Leonard. Mobile robot relocation from echolocation constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):1035–1041, September 2000.
- [5] P. M. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, Australian Centre for Field Robotics. The University of Sydney, March 1999.
- [6] J. Nie, J. Yuh, E. Kardash, and T. Fossen. On-board sensor-based adaptative control of small UUVs in very shallow water. In *Control Applications in Marine Systems*, 1998.
- [7] D. Ribas, J. Neira, P. Ridao, and J.D. Tardós. AUV localization in structured underwater environments using an a priori map. In *Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Lisbon, Portugal, September 2006.
- [8] D. Ribas, P. Ridao, J. Neira, and J.D. Tardós. A method for extracting lines and their uncertainty from acoustic underwater images for SLAM. In *Proceedings of the 6th IFAC Symposium on Intelligent Autonomous Vehicles*, Toulouse, France, September 2007.